StyleSync: Leveraging OpenCV, Siamese Networks, and YOLOv8 for Clothing Image Harmony and Detection

Team Name

E-JUST EleganceNavigators

Team Members

Amro Abdelkhalek Meshref

Ahmed Mohamed Hassan

Ahmed Moustafa Attia

Ahmed Anwar Gad

Table of contents

Table of Figures

Table of Graphs

Overview

Executive Summary:

In the ever-evolving landscape of fashion retail, one persistent challenge faced by consumers is the pursuit of style within budget constraints. Recognizing this, our innovative solution aims to revolutionize fashion shopping by introducing a groundbreaking application that utilizes artificial intelligence to bridge the gap between expensive desires and budget-friendly alternatives.

Problem Statement:

Fashion enthusiasts often encounter the frustration of discovering a desirable piece of clothing, only to be deterred by its high price tag. This hurdle not only affects consumers but also poses a challenge for small retailers seeking to showcase their unique offerings in the vast online marketplace. Additionally, existing image search engines have limitations, focusing mainly on indexed websites and lacking the specificity required for fashion items.

Our Solution:

We developed an application that is designed to empower users with the ability to find affordable alternatives to their favorite fashion items. Our solution uses an AI model that scans through a vast database of clothing available in various stores. Users can simply take a photo of a desired item, and our app will intelligently match it with similar options from a multitude of retailers, including small and local businesses.

Key Features:

1. **AI-Powered Image Recognition:** Our advanced AI model ensures accurate and efficient identification of clothing items from user-uploaded images.
2. **Extensive Database:** We have curated a comprehensive database that includes products from a wide array of retailers, ranging from high-end brands to local boutiques. This ensures users have access to a diverse range of alternatives.
3. **Support for Small Retailers:** By offering a platform for small retailers to showcase their products without the need for a dedicated website, we empower local businesses to reach a broader audience.
4. **Budget-Friendly Recommendations:** Users receive a curated list of budget-friendly alternatives, enabling them to make informed decisions without compromising on style.

5. **No Website Indexing Requirement:** Unlike traditional search engines, our app operates without the need for retailers' websites to be indexed, providing an inclusive platform for all businesses.

Research Background

Leveraging Siamese network-driven image similarity detection, combining it with YOLOv8, and using OpenCV in fashion and e-commerce holds great potential. Siamese networks, a type of computer vision algorithm, allow for intricate feature extraction from clothing items, facilitating nuanced analysis of patterns, colors, and styles. This technology empowers businesses to implement advanced image recognition, enabling users to effortlessly find visually similar clothing items with enhanced precision. The integration of Siamese networks streamlines the shopping experience by providing recommendations and similar clothes. Despite initial challenges with accuracy and variations in image acquisition conditions, Siamese network-powered image similarity detection has made remarkable progress. This advancement can empower fashion platforms to offer a sophisticated and user-friendly interface, enhancing the efficiency of e-commerce processes and satisfying fashion-conscious consumers.

During the search, we found many models that perform similarity detection using Siamese networks, but the thing is it detects all outfit and gets an outfit that may be similar to what the user wants. So, we started by exploring techniques used before and collecting our data.

A paper introduces an innovative method for learning type-aware embeddings in the domain of fashion compatibility, utilizing a Siamese network architecture. It employs an 18-layer Deep Residual Network as the image embedder and integrates visual-semantic embeddings and metric learning to capture both item similarity and compatibility. The model introduces a strategic negative sampling approach, considering item categories, to enhance training effectiveness. Evaluation is conducted on two datasets, the Maryland Polyvore dataset and a proprietary Polyvore dataset, assessing performance through fashion compatibility and fill-in-the-blank tasks. Comparative evaluations against state-of-the-art methods, such as SiameseNet and Compatibility Similarity Network (CSN), highlight the superiority of the proposed approach. The reported accuracy for the model "CSN, T1:1 + VSE + Sim + Metric (512-D)" on the Polyvore Outfits dataset is 86%, indicating a high level of correctness in predicting compatible items and completing fashion recommendations.**[8]**

Another research indeed utilized the FashionAI, DARN, and DeepFashion datasets for attribute-specific fashion retrieval, as there were no existing datasets specifically for this task. The FashionAI dataset, chosen for its high-quality attribute annotations, consists of 180,335 apparel images with hierarchical attribute annotations. DARN, designed for attribute prediction and street-to-shop image retrieval tasks, contains 253,983 images annotated with 9 attributes. In this paper, DeepFashion, a large dataset with various benchmarks, was used for category and attribute prediction. The proposed ASEN model achieved a Mean Average Precision (mAP) of 64.1% across these datasets, showcasing its effectiveness in fine-grained fashion similarity prediction. The study also conducted extensive experiments on model components, loss functions, and weakly-supervised localization, providing a comprehensive analysis of the proposed technique's capabilities.**[2]**

Another paper introduces CSA-Net, an innovative approach for outfit compatibility prediction and complementary item retrieval in fashion. Utilizing a category-based subspace attention network trained with an outfit ranking loss, CSA-Net excels in diverse tasks, outperforming Siamese-Net, Type-aware, and SCE-Net. Notably, CSA-Net achieves a precision of 87% with its outfit ranking loss surpassing triplet loss. The min aggregation function proves superior, enhancing recall@top k in retrieval experiments across fine-grained categories. CSA-Net competes favorably in FITB accuracy and compatibility AUC, even surpassing SCE-Net. Tailored for large-scale retrieval, CSA-Net offers a promising solution for advancing fashion outfit compatibility prediction and complementary item retrieval.**[5]**

Our Approach:

After reviewing the approaches used, we started to make our specific approach which was combining **Siamese network** with **YOLOv8** to enhance precision. Before starting coding, we made our training set then made our database. Next, evaluating positives and negatives. Positives represent similar images and negatives represent dissimilar images. For positive pairs, representing similar images, the network aims to minimize the distance between their embeddings in a shared feature space. This is typically achieved through a contrastive or triplet loss function, ensuring that similar images are mapped closely together. On the other hand, for negative pairs representing dissimilar images, the network is trained to maximize the distance between their embeddings. The objective is to create a margin between dissimilar pairs, pushing them apart in the feature space. Through this process, a Siamese network learns a robust representation where similar inputs are clustered, and dissimilar inputs are distinctly separated

Methodology

YOLO

You Only Look Once (YOLO) is a well-known model architecture and object identification algorithm. Its popularity stems from its use of one of the greatest neural network architectures to provide high accuracy and overall processing speed.

We built an object detection model based on YOLOv8 to be able to identify different genres of clothes shirts, jeans, shoes, and jackets. This will then be used to simplify the search through the database of clothes and to find matching images.

YOLOv8 vs YOLOv5

YOLOv8 and YOLOv5 are both fast object identification models that can process photos in real time. However, YOLOv8 is faster than YOLOv5, making it a superior alternative for real-time object detection applications. When selecting an object detection model, accuracy is a vital issue to consider. YOLOv8 is more accurate in this way than YOLOv5, thanks to various architectural advancements.

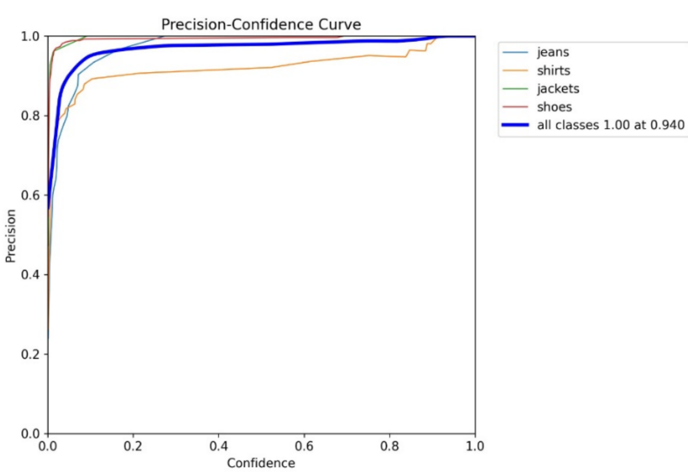
Results

The confusion matrix (as shown in **figure 1**) shows the accuracy of the model against different classifications. It managed to achieve high accuracy in all the 4 classifications while the graph (as shown in **Graph 1**) below shows the precision vs confidence of the model. It shows high area under the curve meaning that it has high precision in detecting objects correctly with higher confidence of what is detected.

Data Transformation:

Several basic processes are carefully carried out in the complex data transformation process for the Siamese model to guarantee the efficient preparation and organization of input data. The first step is to import the necessary packages, which include NumPy, OpenCV, and Matplotlib. These libraries are what you'll need for image processing and visualization jobs later.

After the import of the package, the dataset is arranged into three key arrays: "anchor," "positive," and "negative." These arrays serve as containers, with each one assigned to kinds of images that are essential to the training procedure. Anchor images are stored in the "anchor" array and are essential points of reference for the Siamese model. The "positive" array, which



Graph 1 Precision - Confidence Curve

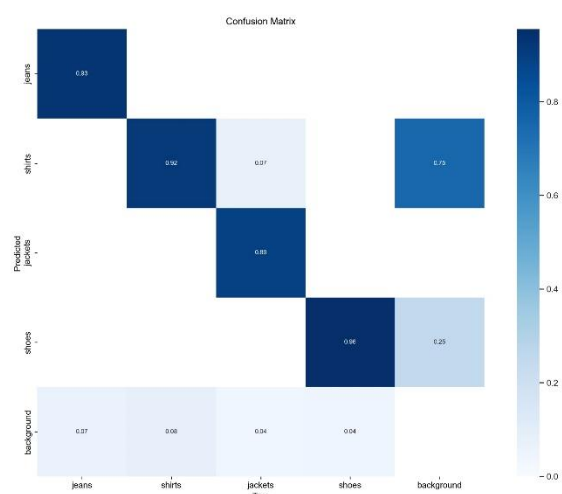


Figure 1 Confusion matrix of different clothes classes

represents comparable things, consists of images coupled with their respective anchor indices at the same time. On the other hand, the "negative" array is made up of pictures that are matched with anchor indices and show different objects in the collection and the implementation is shown in **figure 2**.

```python
# cloth_type: boots, jeans, ets
# training_examples: one triple
# triple: anchor, positive, negative

anc_index = -1

for cloth_type in os.listdir(data_path):
    for training_example in os.listdir(data_path + "/" + cloth_type):
        for triple in os.listdir(data_path + "/" + cloth_type + "/" + training_example):

            if triple == "anchor":
                for image in os.listdir(data_path + "/" + cloth_type + "/" + training_example + "/" + triple):
                    img = cv2.imread(data_path + "/" + cloth_type + "/" + training_example + "/" + triple + "/" +image)
                    anchor.append(img)
                    anc_index += 1
                    del img

            if triple == "positive":
                for image in os.listdir(data_path + "/" + cloth_type + "/" + training_example + "/" + triple):
                    img = cv2.imread(data_path + "/" + cloth_type + "/" + training_example + "/" + triple + "/" +image)
                    positive.append([img,anc_index])
                    del img

            if triple == "negative":
                for image in os.listdir(data_path + "/" + cloth_type + "/" + training_example + "/" + triple):
                    img = cv2.imread(data_path + "/" + cloth_type + "/" + training_example + "/" + triple + "/" +image)
                    negative.append([img,anc_index])
anchor = np.array(anchor)
positive = np.array(positive)
negative = np.array(negative)
```

Figure 2 Siamese Dataset Arrays: Anchor, Positive, Negative

As shown in **figure 2** also, the dataset is carefully extracted after being obtained from preset directories. "Anchor," "positive," and "negative" images are loaded into the appropriate arrays in a methodical manner. The training triplets are made easier by this structured data format. Each triplet consists of an anchor image, a positive image, and a negative image. These triplets provide the fundamental building blocks for the Siamese model's learning, allowing it to successfully discriminate between similar and dissimilar objects.

Furthermore, an important preprocessing step is carried out to maximize the model's effectiveness as shown in **figure 3**. To maintain uniformity in pixel values, images are enlarged to a standard dimension of (105,105,3) and then normalized. By dividing pixel values by 255, normalization is accomplished. This improves the model's overall performance by stabilizing the training process.

```python
def preprocessingData(data):

    try: #for anchor data
        data_resized = []
        for image in data:
            image = cv2.resize(image,image_size) #resizing data into (105,105,3)
            data_resized.append(image)
        data_resized = np.array(data_resized)

        #Normalizing the data
        data_resized = data_resized/255.0;

    except: #for positive or negative data
        data_resized = []
        for img in data:
            image = img[0]/255.0  #Normalizing the data
            anc_index = img[1]
            image = cv2.resize(image,image_size) #resizing data into (105,105,3)
            data_resized.append([image,anc_index])
        data_resized = np.array(data_resized)

    return data_resized
```

Figure 3 Optimizing Input for Enhanced Model Performance

The definition of training and testing datasets is a stage in the next steps. Anchor images are linked with images that are labeled as negative (labelled as 0 to indicate dissimilarity) and with images that are labeled as positive (1 to indicate similarity). This careful data classification creates the basic structure that allows the Siamese model to distinguish between similar and different items in training, we can see what we did in **figure 4**. Visualization is a crucial tool in the effort to validate the efficacy of the data transformation process. Visual examples of validation and anchor images are provided to provide an understanding of the input format that the Siamese model uses. Understanding how the model sees and differentiates between objects that have similarities and

```python
#validation: positive or negative
def addPairsToData(validation, label):
    for img in validation:
        image = img[0]
        anc_index = img[1]
        data.append([anchor_edited[anc_index],image,label])
```

Figure 4 Dataset Validation for Siamese Training those that don't depends on this visual representation, as a result to our code can be seen in **figure 5**.

Lastly, the dataset is split into training and testing sets in an 80:20 ratio to ensure that the model receives an equal amount of data exposure for efficient learning and evaluation. This partitioning helps to create a strong and well-balanced Siamese model that can recognize similar objects based on anchor images by ensuring that the model is trained on a significant percentage of the data while keeping a separate set for rigorous evaluation.

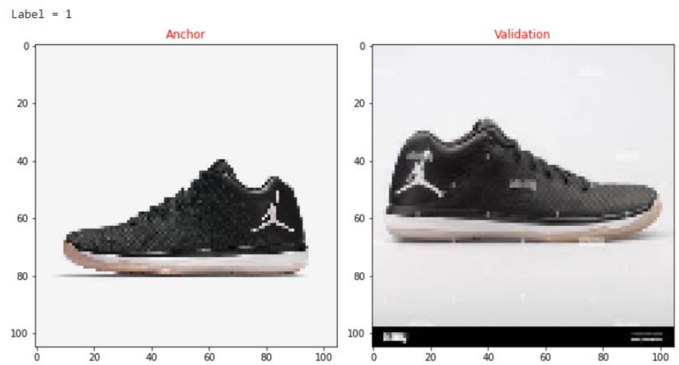


Figure 5 Evaluating Data Transformation: Visual Insights

Siamese network

Siamese networks have become an effective method in computer vision and image analysis for jobs requiring similarity measurement. The underlying idea of Siamese networks is that they can detect the degree of similarity or dissimilarity between two objects by learning to discriminate between pairs of inputs. This is accomplished by employing a shared neural network architecture that processes both inputs concurrently, resulting in a metric space where similarity is indicated by the proximity of representations. Applications for Siamese networks can be found in many different domains, including face recognition and signature verification. They are especially well-suited for jobs where it is crucial to comprehend the similarity between input pairs because of their distinctive design.

How we used it

Within the context of our project, matching clothes is made possible in large part by the use/ of a Siamese network. A Siamese network is particularly good at collecting complex patterns that characterize the essence of various outfits because of the variety of clothing articles and the subtle nature of visual aspects. Here, the goal is to enable users to upload a picture of a specific article of clothing, use YOLOv8 to determine its type, and then use the Siamese network to compare the image of the item with the wide variety of apparel that is kept in our database.

In order to train our Siamese network, we collected a dataset from a GitHub repository that included several kinds of clothes, including shirts, jeans, jackets, and shoes. We have meticulously selected three image triplets from this dataset: an anchor image that represents the item that was searched, a positive image that shows the same kind of type of clothing, and a negative image that shows a different kind of clothing. Through careful selection, the network was trained to distinguish between clothing items that are similar and those that are not, which will enable precise matching while searching.

Architecture and Implementation

The research paper "Siamese Neural Networks for One-shot Image Recognition" by Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov **[4]** served as our model for creating the design of our Siamese neural network. We have carefully crafted our architecture to tackle the unique problems presented by one-shot picture identification, giving each layer a specialized function to improve the network's capacity to identify and match clothing items.

The core of our system is the use of convolutional layers. These layers are excellent at collecting hierarchical information in photos, which is exactly what's required to identify subtle patterns in apparel. The choice to extract different features while increasing computing efficiency led to the usage of various filters with a channel size as a multiple of 16. This deliberate balancing makes sure the network can accurately identify and depict a large variety of apparel attributes.

The introduction of non-linearity into the model is mostly dependent on activation functions. The first layers' Rectified Linear Unit (ReLU) activation makes it easier to identify intricate correlations in the data. As the network develops, the requirement to produce outputs that resemble probabilities drives the switch to sigmoidal activation, which is essential for assessing the degree of similarity across clothing pieces.

In order to focus on the most important characteristics and reduce computational complexity, feature maps are deliberately down-sampled using max-pooling. Selecting a stride of two and a filter size of two results in a compromise between decreasing computational load and maintaining important information.

A thorough grasp of garment attributes is provided by the network's ability to recognize global dependencies in the data thanks to the inclusion of fully connected layers. The choice to combine the units of the last convolutional layer into a single vector guarantees a smooth transfer of spatial data to be processed further.

When assessing how similar two pieces of clothing are to one another, the layer that calculates the induced distance measure between Siamese twins is crucial. Given the nature of our matching job, the application of a sigmoidal activation function yields similarity scores ranging from 0 to 1.

Most importantly, the addition of learned parameters (αj) makes the model more flexible. These parameters enable the network to prioritize significant characteristics according to their contribution to overall similarity by dynamically weighing the significance of component-wise distances.

The architecture that is shown in **figure 6** represents the result of these careful design decisions. It turned out to be the best setup for our verification work, demonstrating how well our method handled the complexities of one-shot image recognition. This architecture's compatibility

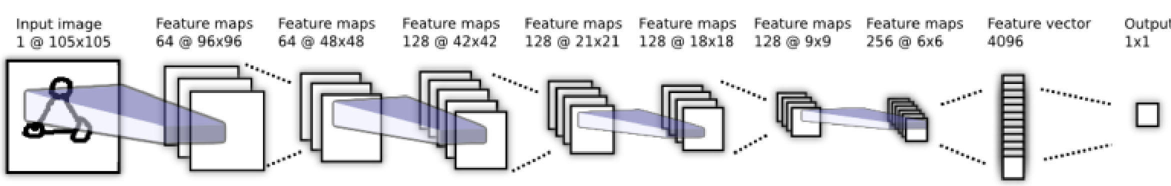


Figure 6 Siamese network architecture design

and effectiveness for our project's goals are demonstrated by how well it matches our clothes matching application.

Accuracy Metrics:

Our trained Siamese network exhibits commendable performance, achieving a training accuracy of 100%. This high accuracy demonstrates the network's proficiency in learning the intricate features that distinguish different clothing items. Furthermore, during testing, the network maintains a robust accuracy of 98.5%, showcasing its capability to generalize well to unseen data. This level of accuracy instills confidence in the effectiveness of our Siamese network for clothing matching in real-world scenarios.

Application:

To create the program, the kivy library was used. It is a library used to create apps and games, but we used it to make a good and simple look for users. The program is divided into 2 parts.

First part is for user, it enables user to take a photo of the product or clothes he searches for, and then displays the result.

Second part is for the store, it provide store with option to take photos of the products and upload them so, the user can get results of similar photos from the store with the help of our model.

Conclusion

To sum-up, our image-matching application represents a breakthrough in addressing challenges faced by both consumers and small retailers in the online fashion industry. Leveraging cutting-edge technologies like Siamese networks and YOLOv8 with the help of use of OpenCV, our solution provides users with precise and affordable alternatives to desired fashion items.
The Siamese network showcased exceptional accuracy, achieving **100% during training** and an **impressive 98.5% during testing**. Combined with YOLOv8 for efficient object detection, our application streamlines the search process and supports small retailers in reaching a broader audience.
Our commitment to robust data transformation processes ensures a user-friendly interface and efficient model training. The implementation, using the kivy library, enhances the user experience, offering a simple platform for both users and stores. Our application, backed by state-of-the-art technologies, stands as a great solution in fashion matching and online retail.

Our work

Our Project can be found on the following respiratory:
https://github.com/amromeshref/SimilarClothes

References

[1] Aflahi, T. F. (2019, April 7). *Improving siamese network performance*. Medium. https://medium.com/@kuzuryu71/improving-siamese-network-performance-f7c2371bdc1e

[2] Dong, J., Ma, Z., Mao, X., Yang, X., He, Y., Hong, R., & Ji, S. (2021). Fine-Grained Fashion Similarity Prediction by Attribute-Specific Embedding Learning. *IEEE Transactions on Image Processing, 30*, 8410–8425. https://doi.org/10.1109/TIP.2021.3115658

[3] Gupta, P. (2020, November 5). *Illustrated guide to siamese network*. Medium. https://towardsdatascience.com/illustrated-guide-to-siamese-network-3939da1b0c9d

[4] Koch, G., Zemel, R., & Salakhutdinov, R. (2015, July). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop* (Vol. 2, No. 1).

[5] Lin, Y.-L., Tran, S., & Davis, L. S. (2020). Fashion Outfit Complementary Item Retrieval. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr42600.2020.00337

[6] Nag, R. (2022, November 19). *A comprehensive guide to siamese neural networks*. Medium. https://medium.com/@rinkinag24/a-comprehensive-guide-to-siamese-neural-networks-3358658c0513

[7] Rosebrock, A. (2023, June 9). *Siamese networks with Keras, tensorflow, and Deep Learning*. PyImageSearch. https://pyimagesearch.com/2020/11/30/siamese-networks-with-keras-tensorflow-and-deep-learning/

[8]Vasileva, M. I., Plummer, B. A., Dusad, K., Rajpal, S., Kumar, R., & Forsyth, D. (2018). Learning Type-Aware Embeddings for Fashion Compatibility. arXiv preprint arXiv:1803.09196. https://openaccess.thecvf.com/content_ECCV_2018/papers/Mariya_Vasileva_Learning_Type-Aware_Embeddings_ECCV_2018_paper.pdf

[9] Yamazaki, Kimitoshi. (2017). *A method of classifying crumpled clothing based on image features.* derived from clothing fabrics and wrinkles. Autonomous Robots. 41. 10.1007/s10514-016-9559-z.