**IMAGE SIMILARITY MODEL WITH SIAMESE NETWORK**

By

## SAGNIK GHOSH

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| **TITLE** | **PAGE NO.** |
| **LIST OF FIGURES AND TABLES** | I |
| **DECLARATION** | II |
| **CERTIFICATE** | III |
| **ACKNOWLEDGEMENT** | IV |
| **ABSTRACT** | V |
| **CHAPTER 1 – INTRODUCTION** | 1-4 |
| **CHAPTER 2 - FEASIBILITY STUDY,**  **REQUIREMENTS ANALYSIS &**  **DESIGN** | 5-10 |
| **CHAPTER – 3 IMPLEMENTATION** | 11-17 |
| **CHAPTER 4 - RESULTS** | 18-22 |
| **CHAPTER 5- CONCLUSION** | 23-25 |
| **REFERENCES** | 26 |

**LIST OF FIGURES AND TABLES**

**Fig 2.1: Problem Design**

**Table 2.1: Literature Survey**

**Fig 3.1: Flow chart**

**Fig 3.2: Setting up colab environment**

**Fig 3.3: Importing Dataset**

**Fig 3.4: Organize and Load Data**

**Fig 3.5: Batch Generation**

**Fig 3.6: Validate Data**

**Fig 3.7: Feature Generation**

**Fig 3.8: Define and train Siamese Model**

**Fig 4.1: Visual Mode Feedback**

**Fig 4.2: T-shirt vs Ankle-Boot Plot**

**Fig 4.3: Neighbor Visualization**

I

**DECLARATION**

I hereby declare that this project has been done by me under the supervision of Mr. Arvind Kumar, Assistant Professor, JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

**SUPERVISED BY:**

**Mr. Arvind Kumar**

**Assistant Professor (Senior Grade)**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

**SUBMITTED BY:**

## SAGNIK GHOSH (201295)

KSHITIZ BASHYAL (201306)

COMPUTER SCIENCE & ENGINEERING DEPARTMENT

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

II

**CERTIFICATE**

This is to certify that the work which is being presented in the project report titled “IMAGE SIMILARITY MODEL” in partial fulfilment of the requirements for the award of the degree of B. Tech in Computer Science & Engineering and submitted to the department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by SAGNIK GHOSH (201295) KSHITIZ BASHYAL (201306), during the period from January 2022 to May 2022 under the supervision of Mr. Arvind Kumar, Department of Computer Science and Engineering, Jaypee University of Technology, Waknaghat.

**SAGNIK GHOSH (201295)**

**KSHITIZ BASHYAL (201306)**

The above statement made is correct to the best of my knowledge.

**Mr. ARVIND KUMAR**

**Assistant Professor (Senior Grade)**

**COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY**

**JAYPEE UNIVERSITY OF TECHNOLOGY, WAKNAGHAT**

III

**ACKNOWLEDGEMENT**

First of all, I would like to express my heartiest gratitude and gratefulness to almighty god for his divine blessings helped us to complete the project work successfully in the given period of time.

I extend my heartiest thanks to our project Supervisor Mr. Arvind Kumar, Assistant Professor (SG), Department of Computer Science & Engineering, Jaypee University of Information Technology, Waknaghat. Vast knowledge and keen interest of our supervisor in the field of Machine Learning helped us a lot to execute this project. Her endless patience, scholarly guidance, continual encouragement, constant, energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all the stages have made it possible to complete this project.

I would also generously thank each one of those individuals who have helped me directly or indirectly in successfully carrying out the execution of this project. In this situation, I also want to thank the various staff individuals, both educating and non- instructing, which have developed their convenient help and facilitated my undertaking.

Last but not the least, I must acknowledge with due respect the constant support and faith of my parents.

**SAGNIK GHOSH (201295)**

**KSHITIZ BASHYAL (201306)**

IV

**ABSTRACT**

An image similarity model is a type of machine learning algorithm that is used to compare and measure the similarity between two or more images. The model works by extracting various features from the images, such as color, texture, and shape, and then using a distance metric to calculate the similarity between them. These models have a wide range of applications in various fields, including content-based image retrieval, image classification, and object recognition.

Object recognition is another area where image similarity models are commonly used. In this context, the model is trained to identify specific objects or features in an image. For example, an image similarity model can be trained to recognize faces, which can be useful in security and surveillance applications.

There are various approaches to building an image similarity model, including traditional machine learning techniques such as k-nearest neighbors and support vector machines, as well as deep learning methods such as convolutional neural networks. Here we are going to use “Siamese Network” Deep learning models have become increasingly popular in recent years, as they are able to extract more complex features from images and achieve higher accuracy in image similarity tasks.

Overall, image similarity models are a powerful tool for comparing and measuring the similarity between images. With their wide range of applications and the increasing availability of large image datasets, these models are likely to play an increasingly important role in many fields in the years to come.

V

**CHAPTER – 1 INTRODUCTION**

### **IMAGE SIMILARITY**

Image similarity refers to the degree of similarity or dissimilarity between two or more images. It is a concept used in many fields, including computer vision, image processing, and machine learning.

To measure image similarity, various features of the images are analyzed and compared. These features can include color histograms, texture, edges, and shape. The similarity measure can be calculated using various methods, such as Euclidean distance, cosine similarity, or correlation coefficient.

Image similarity has many practical applications. In content-based image retrieval, for example, it is used to search for images similar to a given query image. This can be useful in various contexts, such as finding similar images for product recommendations or searching for specific images in a large collection.

### MACHINE LEARNING

MACHINE LEARNING is a part of man-made brainpower (Artificial Intelligence) and software engineering which centers around the utilization of information and calculations to impersonate the way that people learn, progressively working on its precision. It is significant in light of the fact that it provides undertakings with a perspective on patterns in client conduct and business functional examples, as well as supports the improvement of new items. It has turned into a critical cutthroat differentiator for some organizations. It uses libraries, inbuilt functions, models to produce the desired result. It uses a collection of data, on which it is trained and later on used to produce the results for which we trained our model. Python libraries help its user to easily and hastily perform the task without writing the code from scratch unlike other languages. Some of the important libraries of python are pandas, numpy, seaborn, matplotlib etc. The models/classifiers are also a key part of machine learning, they are used to train the model in different ways, some classifiers are KNN, logistic regression, linear regression etc. These models have different functionalities and different ways in which they are operated. We shall discuss the models and libraries further in the document according to their use in the project.

### OBJECTIVE

The objective of image similarity using a Siamese network is to learn a representation of images in which similar images are mapped close together in a low-dimensional embedding space, while dissimilar images are mapped farther apart. The goal is to capture the inherent similarity or dissimilarity between images based on their visual content.

The Siamese network architecture is designed to compare and measure the similarity between two input images. By training the Siamese network on pairs of images with corresponding labels indicating their similarity, the network learns to extract meaningful features and embeddings that capture the similarities and differences between images.

The ultimate objective is to create a model that can accurately assess the similarity or dissimilarity between two images, even when dealing with unseen images. This learned representation can be used for various tasks, such as image retrieval, image clustering, content-based image search, or even as a component in more complex systems like face recognition or object detection.

In summary, the objective of image similarity using a Siamese network is to learn a representation that effectively captures and measures the similarity between images, enabling various applications that rely on comparing and matching images based on their visual content.

### MOTIVATION

The motivation for building an image similarity model comes from the growing amount of visual data available today, from social media platforms to scientific research databases. With the rapid increase in digital images, there is a growing need to develop techniques that can effectively analyze and process visual data.

Image similarity models provide a way to analyze and compare images based on their visual features, allowing for efficient image retrieval, object recognition, and image clustering. This can have a wide range of applications, including e-commerce, social media, medical imaging, and scientific research.

For example, in e-commerce, image similarity models can be used to recommend products to customers based on their visual preferences. In social media, they can be used to identify similar images for content moderation or to group related images together. In medical imaging, they can help doctors diagnose diseases based on visual similarities with other medical images. In scientific research, they can aid in data analysis and classification based on visual features.

Overall, the motivation for building an image similarity model is to enable efficient and effective analysis of visual data, which can have significant practical applications in various fields.

### LANGUAGE USED

Python programming language is used in the scripting of this project. It provides a large no. of libraries which are most commonly used in building up a machine learning project.

* KERAS: Keras is used when a neural network needs to be quickly built and executed in a few lines of code and is a wrapper for TensorFlow. It contains implementations of commonly used neural network elements such as layers, targets, activation functions, optimizers, and tools to make working with images and text data easier.
* TENSORFLOW: TensorFlow is the library responsible for the mathematical computation behind a neural network which includes computing and optimizing the loss function by adjusting the weights and biases in order to get accurate prediction results.
* NUMPY: NumPy stands for Numerical Python. It is a Python library utilized for working with arrays. It likewise has capacities for working in space of direct polynomial math, Fourier change, and grids.
* PANDAS: Pandas is python library used to analyze the data and manipulate it. It helps to manipulate data using data structures which are provided by the library itself, like list, data frames etc.
* MATPLOTLIB: It is python library which assists users to plot their understandings of the data, i.e., visualization of the data in various forms, like bar, histogram, scatter plot, pie chart etc.
* DATASET: It is a collection of set of information, which is composed of separate elements which can be manipulated as one by a computer.

**CHAPTER - 2 FEASIBILITY STUDY, LITERATURE SURVEY & REQUIREMENTS ANALYSIS**

### **2.1 FEASIBILITY STUDY**

A feasibility study for an image similarity model involves evaluating the technical, economic, and operational aspects of developing and deploying such a model. Here are some considerations for each of these aspects:

**Technical feasibility:**

- Availability of image datasets for training and validation

- Adequate computing resources for training and running the model

- Ability to extract relevant features from the images and accurately measure their similarity

- Availability of appropriate algorithms and libraries for implementing the model

- Compatibility of the model with existing infrastructure and systems

**Economic feasibility:**

- Estimated cost of development, including time, resources, and equipment

- Potential return on investment, such as increased efficiency or revenue from improved image search capabilities

- Comparison of the cost of developing an image similarity model in-house versus outsourcing or using existing software

**Operational feasibility:**

- Availability of skilled personnel to develop, maintain, and operate the model

- Compatibility with existing business processes and workflows

- Integration with existing software and hardware systems

- Potential impact on end-users, such as ease of use and accuracy of results

Based on these considerations, a feasibility study for an image similarity model could involve conducting a cost-benefit analysis, evaluating the availability and quality of image datasets, assessing the technical capabilities of the development team, and analyzing the potential impact on end-users and business processes. The results of the study can help inform the decision to proceed with developing an image similarity model and guide the planning and implementation of the project.

**PROBLEM DESIGN**

PROBLEM STATEMENT

ANALYSING THE REQUIREMENTS

OBTAINING DATASET

ANALYSING DATASET

SPLITTING THE DATASET

TRAINING THE MODEL

RESULTS

**Fig 2.1: Problem Design**

### 2.2 LITERATURE SURVEY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SNO.** | **NAME** | **AUTHOR** | **YEAR PUBLISHED** | **WORK DONE** |
| **1** | Deep Metric Learning for Visual Search and Image Retrieval | Radenovic | 2018 | The model learns a similarity metric that maps images to a low-dimensional embedding space, where similar images are closer together. |
| **2** | Multi-Similarity Loss with General Pair Weighting for Deep Metric Learning | Wang | 2019 | This paper proposes a multi-similarity loss function for deep metric learning, which incorporates both global and local similarities between images. |
| **3** | Improved Deep Metric Learning with Multi-class N-pair Loss Objective | Chen | 2020 | This paper proposes a multi-class N-pair loss function for deep metric learning, which aims to maximize the margin between similar and dissimilar image pairs. |
| **4.** | EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks | Tan and Le | 2019 | This paper proposes a new scaling method for convolutional neural networks, which achieves better accuracy and efficiency compared to previous methods. |
| **5.** | Learning to Compare: Relation Network for Few-Shot Learning | Sung | 2018 | This paper proposes a relation network for few-shot learning, which learns to compare images and classify them based on their similarity. |

**Table 2.1: Literature Survey**

### 2.3 REQUIREMENTS ANALYSIS

Siamese networks are commonly used for image similarity tasks. They are a type of neural network architecture that learns to identify similarities and differences between pairs of images. To train a siamese network for image similarity, the following requirements should be considered:

HARDWARE REQUIREMENTS

1. **CPU**: A multi-core CPU is required for efficient data processing and feature extraction. A CPU with a clock speed of at least 2.5 GHz and 8 cores is recommended.

2. **RAM**: The amount of RAM required depends on the size of the dataset and the complexity of the model. For small to medium-sized datasets, 16-32 GB of RAM is usually sufficient. For larger datasets, 64 GB or more may be required.

3. **GPU**: A GPU is recommended for faster model training and inference. A GPU with at least 8 GB of VRAM is recommended, and multiple GPUs can be used in parallel for faster processing. NVIDIA GPUs are the most commonly used for deep learning applications.

4. **Storage**: A high-speed SSD or NVMe drive is recommended for storing the dataset and model files. A storage capacity of at least 1 TB is recommended for large datasets.

5. **Power Supply**: A high-quality power supply with sufficient wattage is required to power the CPU, GPU, and other components. A power supply with a wattage of at least 600W is recommended.

SOFTWARE REQUIREMENTS

• Operating System: Windows 11.

• Platform: Google colaboratory

• Programming Language: Python

DATASET

The dataset is used from Kaggle. The dataset is “Fashion Product Images Dataset-MNIST”.

Fashion-MNIST is a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. Zalando intends Fashion-MNIST to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits.The dataset contains professionally shot high resolution product images, and have multiple label attributes describing the product which was manually entered while cataloging.

**CHAPTER 3 – IMPLEMENTATION**

### 3.1 FLOW CHART

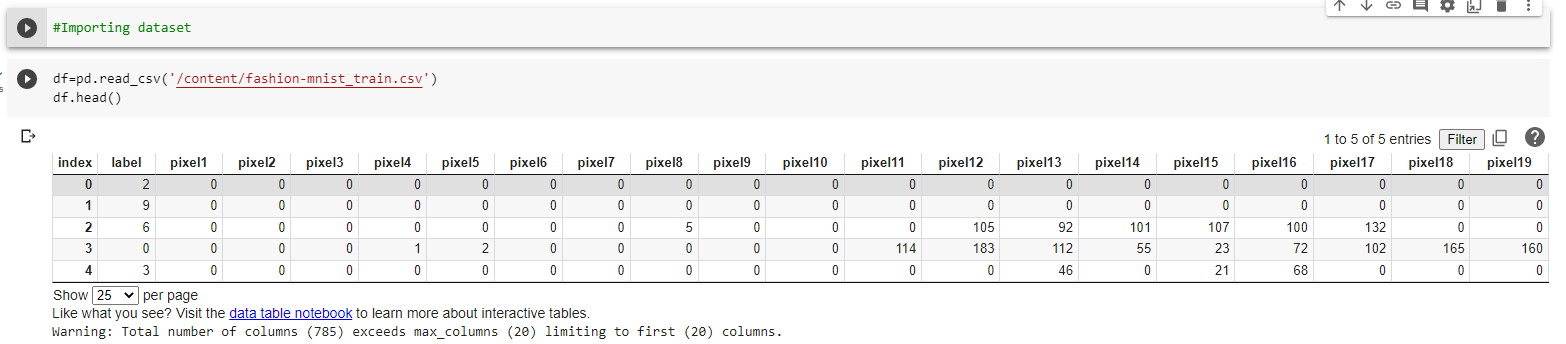
**Fig 3.1: Flow chart**

· Setting up the colab environment by importing libraries.



**Fig 3.2: Setting up colab environment**

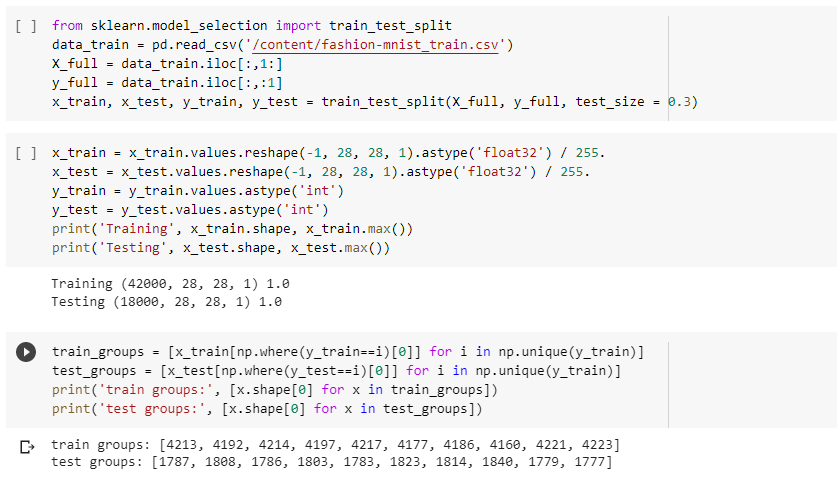
* Importing the dataset



**Fig 3.3: Importing Dataset**

* Preprocessing

LOAD AND ORGANIZE DATA



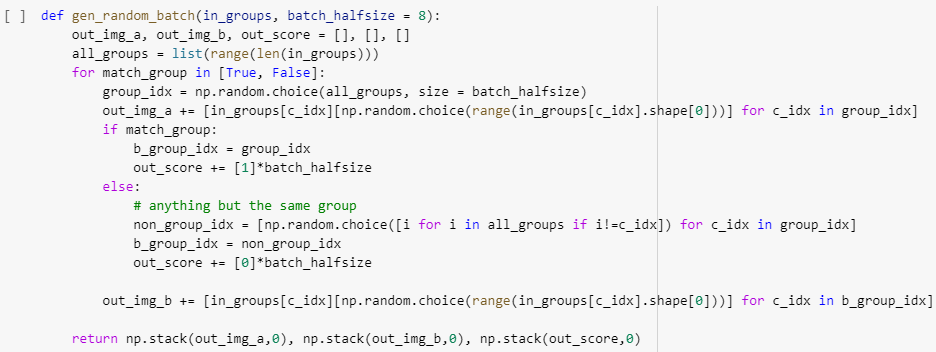
**Fig 3.4: Organize and Load Data**

1. It imports the necessary libraries and modules.
2. It reads a fashion training dataset from a CSV file using Pandas.
3. It splits the dataset into input features (X) and target variable (y).
4. It further splits the data into training and testing sets using the **train\_test\_split** function from scikit-learn.
5. It reshapes the input feature data into a 4-dimensional array to match the expected input shape of a convolutional neural network (CNN) model.
6. It normalizes the input feature data by scaling the pixel values between 0 and 1.
7. It converts the target variable data to integer format.
8. It prints the shape and maximum value of the training and testing data arrays.
9. It groups the training and testing data subsets by their corresponding fashion class labels.
10. It prints the number of samples in each group for both training and testing data.

**BATCH GENERATION**

Here the idea is to make useable batches for training the network. We need to create parallel inputs for the A and B images where the output is the distance. Here we make the naive assumption that if images are in the same group the similarity is 1 otherwise it is 0.

If we randomly selected all of the images we would likely end up with most images in different groups.



**Fig 3.5: Batch Generation**

**Validate Data**

Here we make sure the generator is doing something sensible, we show the images and their similarity percentage.

****

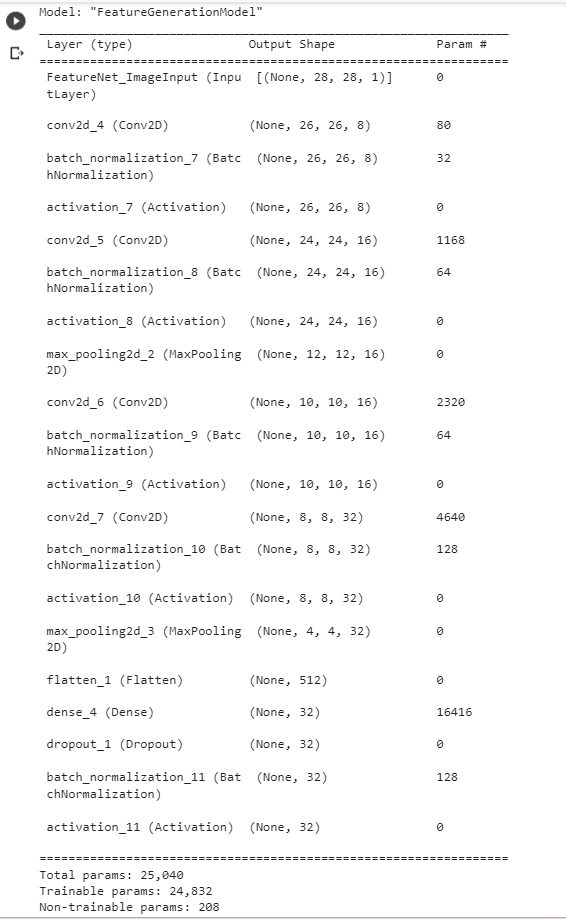
**Fig 3.6: Validate Data**

**Feature Generation**

Here we make the feature generation network to process images into features. The network starts off randomly initialized and will be trained to generate useful vector features from input images

The following code defines a feature extraction model with convolutional and pooling layers, followed by fully connected layers. The model takes the input images and produces a feature vector as output, which can be used for further analysis or as input to another model.

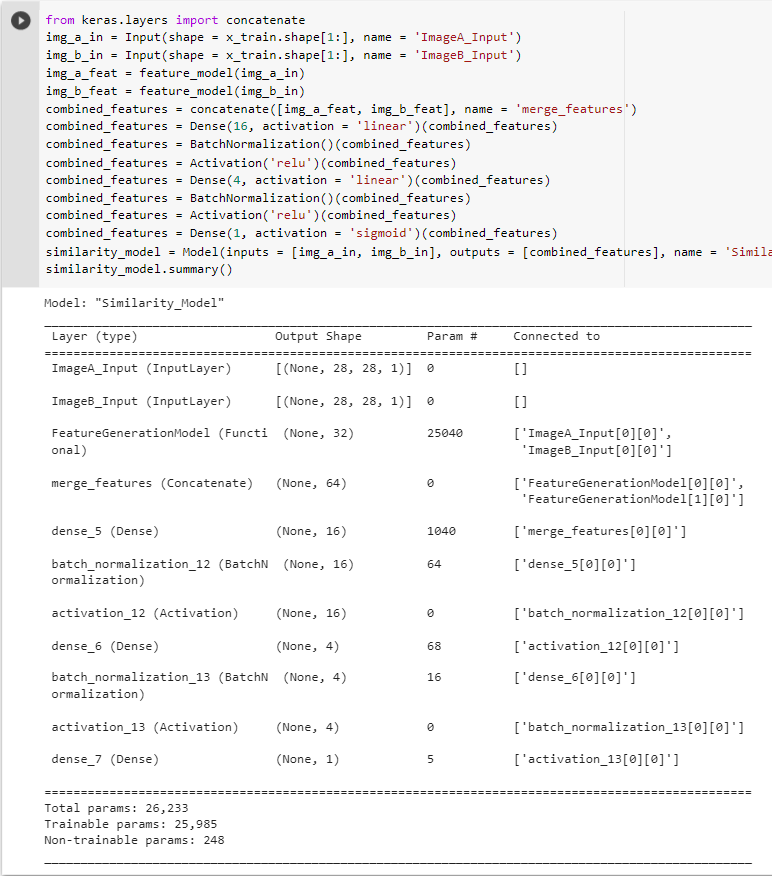




**Fig 3.7: Feature Generation**

* Define and Train Siamese Model

We apply the feature generating model to both images and then combine them together to predict if they are similar or not. The model is designed to very simple. The ultimate idea is when a new image is taken that a feature vector can be calculated for it using the FeatureGenerationModel. All existing images have been pre-calculated and stored in a database of feature vectors. The model can be applied using a few vector additions and multiplications to determine the most similar images. These operations can be implemented as a stored procedure or similar task inside the database itself since they do not require an entire deep learning framework to run.

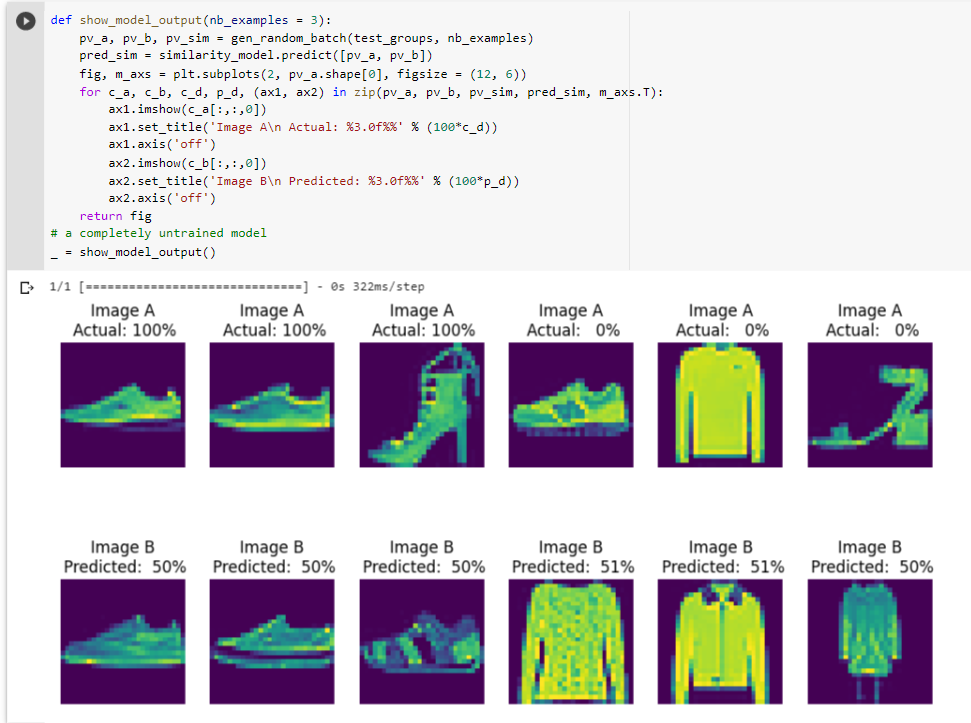


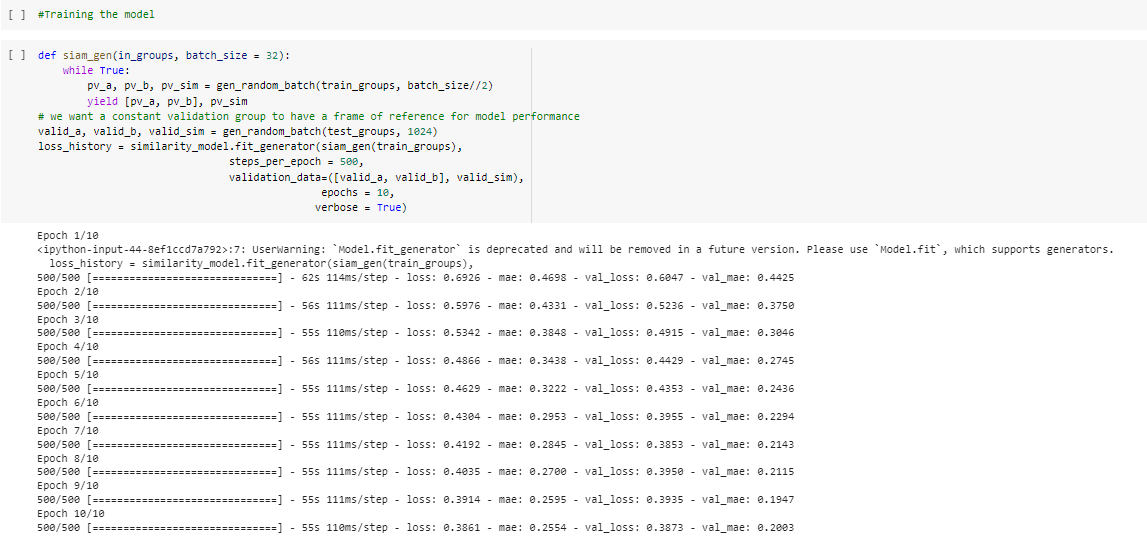
**Fig 3.8: Define and train Siamese Model**

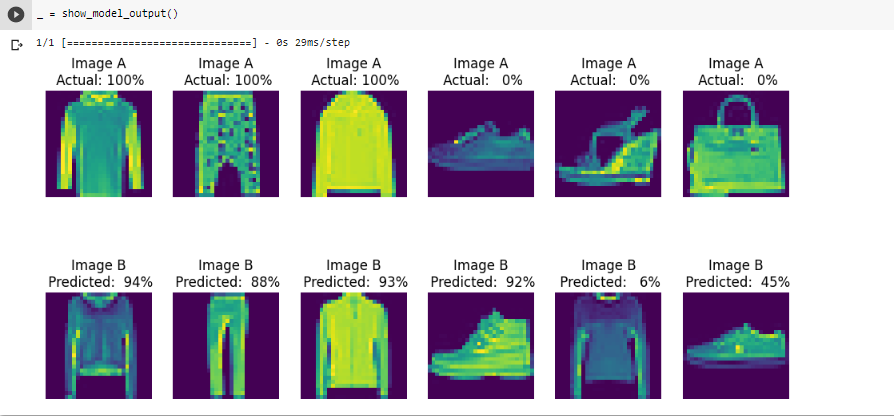
**CHAPTER 4 – RESULTS**

* Visual Mode Feedback

Here we visualize what the model does by taking a small sample of randomly selected A and B images the first half from the same category and the second from different categories. We then show the actual distance (0 for the same category and 1 for different categories) as well as the model predicted distance. The first run here is with a completely untrained network so we do not expect meaningful results.





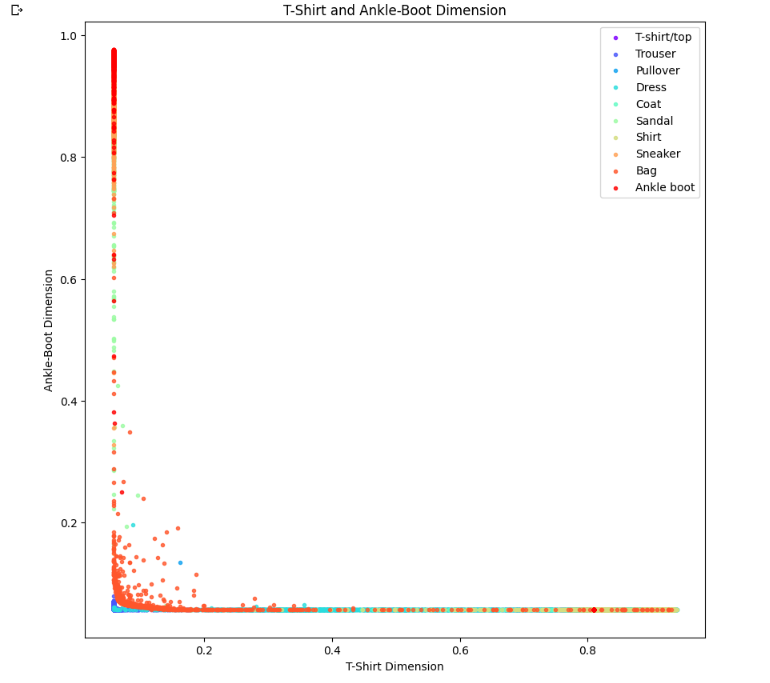


**Fig 4.1: Visual Mode Feedback**

**T-Shirt vs Ankle Boot Plot**

Here we take an random t-shirt and ankle boot (categories 0 and 9) images and calculate the distance using our network to the other images

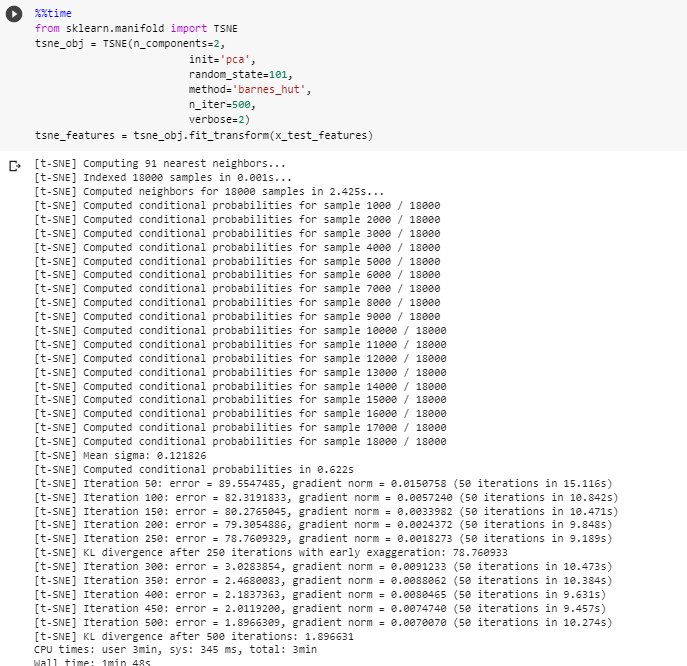


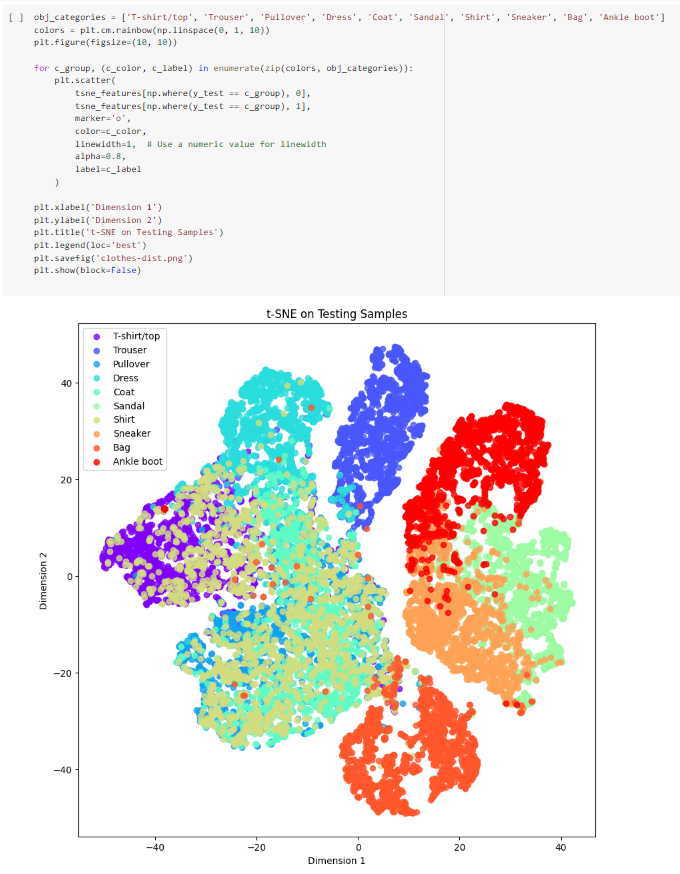


**Fig 4.2: T-shirt vs Ankle-Boot Plot**

**Neighbor Visualization**

For this we use the TSNE neighborhood embedding to visualize the features on a 2D plane and see if it roughly corresponds to the groups. We use the test data for this example as well since the training has been contaminated





**Fig 4.3: Neighbor Visualization**

**CHAPTER 5 – CONCLUSION**

**4.1 Discussion on the results achieved**

1. Improved Similarity Metric: Siamese networks are effective in learning a similarity metric that can capture meaningful relationships between images. By training the network on pairs of similar and dissimilar images, it learns to map similar images closer together and dissimilar images farther apart in the embedding space. This learned similarity metric can lead to more accurate and robust image comparisons.
2. Enhanced Image Retrieval: Siamese networks have shown improved performance in image retrieval tasks. By using the learned embeddings, similar images can be retrieved more accurately and efficiently. This is particularly useful in scenarios such as content-based image search, where finding images with similar visual content is the objective.
3. Robustness to Image Variations: Siamese networks have demonstrated robustness to variations in lighting conditions, viewpoints, and image transformations. The shared weights and architecture of Siamese networks allow them to extract features that are invariant to these variations, resulting in more reliable and accurate image similarity comparisons.

**4.2 Application of the Minor Project**

1. **Content-Based Image Retrieval:** Siamese networks can be used to build systems for content-based image retrieval. Given a query image, the network can compare it with a large database of images and retrieve visually similar images based on their learned embeddings. This is useful in applications like image search engines, recommendation systems, and image organization.
2. **Image Clustering:** Siamese networks can be utilized for image clustering, where similar images are grouped together. By leveraging the learned embeddings, images with similar visual content can be identified and clustered. This is useful for organizing large image datasets, exploring image collections, and discovering visual patterns.
3. **Image Verification:** Siamese networks can be employed for image verification tasks, such as counterfeit detection, forgery detection, or document authentication. The network can compare pairs of images and determine if they are genuine or manipulated based on their learned similarity metric.
4. **Anomaly Detection:** Siamese networks can be used for anomaly detection in images. By learning the normal representation of a particular scene or object, the network can identify anomalies or outliers that deviate from the learned patterns. This is useful in applications such as surveillance, quality control, and medical imaging.
5. **Image Restoration:** Siamese networks can assist in image restoration tasks by comparing corrupted or degraded images with their corresponding clean or high-quality versions. The network can learn to generate restored images by capturing the similarities and differences between the input and reference images.

**4.3 Limitation of the Minor Project**

While image similarity with Siamese networks has several advantages, it also has certain limitations. Here are some of the limitations of using Siamese networks for image similarity:

1. **Training Data Requirements:** Siamese networks require a large amount of labeled training data, consisting of pairs of similar and dissimilar images. Acquiring such labeled data can be time-consuming and expensive, especially for specialized or domain-specific image similarity tasks.
2. **Computational Complexity:** Siamese networks can be computationally expensive to train and evaluate, especially when dealing with large-scale datasets. The pairwise comparison nature of Siamese networks requires multiple forward passes for each pair of images, resulting in increased computational complexity.
3. **Sensitivity to Pair Sampling:** The performance of Siamese networks can be sensitive to the sampling strategy used to create pairs of similar and dissimilar images during training. Choosing an appropriate sampling strategy is crucial to ensure that the network learns meaningful and representative image similarities.

**4.4 Future Work**

After reading Research papers for the specifics mentioned above, we discovered that there is still lot more left for us to work, some of the points are given below:

1. **Advanced Training Strategies:** Investigating advanced training strategies can enhance the learning capabilities of Siamese networks. This may involve incorporating self-supervised learning techniques, adversarial training, or curriculum learning to improve the discrimination power of the network.
2. **Fine-Grained Image Similarity:** Fine-grained image similarity focuses on identifying similarities and differences between images of the same category with subtle variations. Future work can explore techniques to capture fine-grained details and improve the discrimination ability of Siamese networks in such scenarios.
3. **Large-Scale Image Similarity:** Scaling up Siamese networks to handle large-scale image datasets is an important research direction. Developing strategies to efficiently process and learn from massive amounts of image data can help in achieving better performance and scalability.
4. **Real-Time and Efficient Implementations:** Developing real-time and efficient implementations of Siamese networks for image similarity is another important area. Exploring hardware acceleration, model compression techniques, and network optimization methods can improve the deployment and practicality of image similarity systems.

**REFERENCES**

The following are the references that were considered for this project –

1. Hadsell, R., Chopra, S., & LeCun, Y. (2006). *Dimensionality reduction by learning an invariant mapping. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (Vol. 2, pp. 1735-1742). IEEE.
2. Koch, G., Zemel, R., & Salakhutdinov, R. (2015). *Siamese neural networks for one-shot image recognition. In Proceedings of the 32nd International Conference on Machine Learning (ICML)* (Vol. 37, pp. 956-963). JMLR Workshop and Conference Proceedings.
3. Chopra, S., Hadsell, R., & LeCun, Y. (2005). *Learning a similarity metric discriminatively, with application to face verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (Vol. 1, pp. 539-546). IEEE.
4. Zagoruyko, S., & Komodakis, N. (2015). *Learning to compare image patches via convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4353-4361). IEEE.