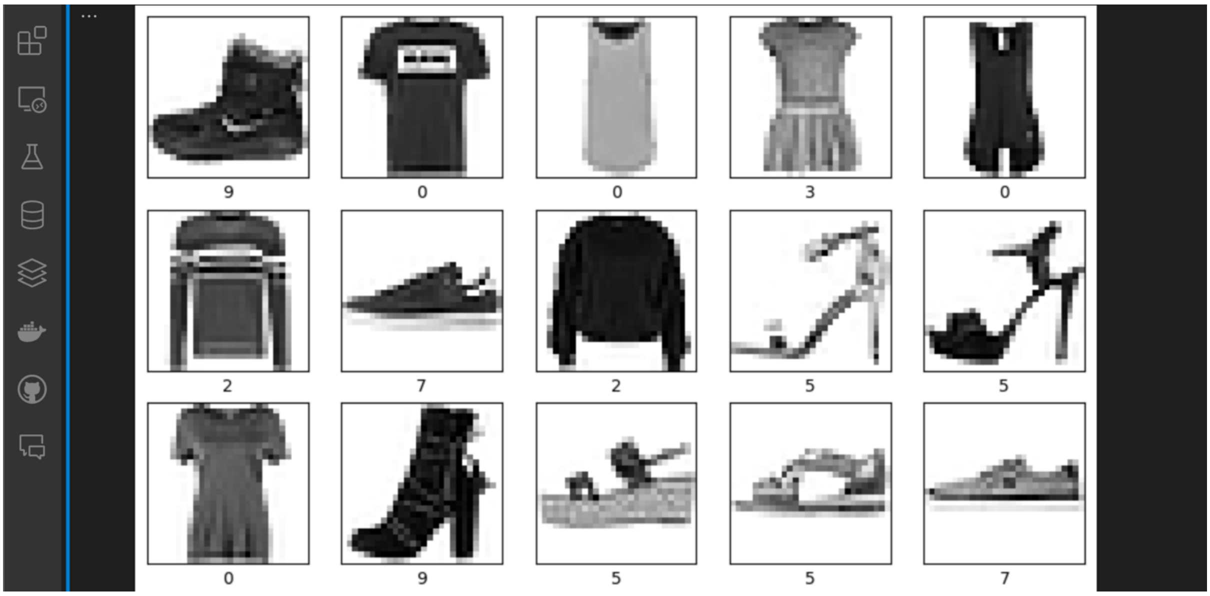
Building a Google Lens alternative...

ojasva963@gmail.com - Ojasva Singh - 7838067886

Objective: To develop an alternative to Google Lens by implementing image similarity using multiple approaches.

Dataset: I decided to use the fashion-MNIST dataset available on **KAGGLE** since **shoppin** uses similar images.



Methods evaluated:

1. Deep Metric Learning (Siamese Network)
2. Hashing Based Method
3. Autoencoder-Based Latent Space Mapping
4. Custom CNN feature extraction
5. Variational Autoencoder

Evaluation Metrics:

1. **Precision@K –** It is the proportion of relevant instances among the retrieved instances.

2. **Recall@K –** Proportion of relevant instances retrieved over the total number of instances

3. **Retrieval Accuracy** – Percentage of queries where the correct label is among the top K results.

4. **Search Time –** Average time taken for each query to perform similarity search.

The following table was obtained after evaluating metrics for each method.

	Precision@K	Recall@K	Retrieval Accuracy	Search Time
Siamese Network	0.568	0.00047333333333333336	0.84	0.011797528266906738
Hashing-Based Method	0.29399999999999993	0.00024500000000000005	0.82	0.002420353889465332
Autoencoder	0.826	0.0006883333333333335	0.96	0.36187545776367186
Custom CNN	0.9080000000000001	0.0007566666666666669	0.96	0.012512509822845458
Variational Autoencoder	0.816	0.0006800000000000004	0.96	0.008396618366241455

Detailed Analysis of each method:

1. **Siamese Network –**
    a. The values of Precision@K and Recall@K are moderate. These can be treated as the baseline.
    b. The search time is relatively low which makes this method suitable for applications requiring moderate speed and accuracy.
    c. Requires more training data to improve performance.
    d. **Pros: Moderate search time.**
    e. **Cons: Low Precision and Recall**

2. **Hashing-Based** -
    a. Very low values of Precision@K and Recall@K.
    b. Has the fastest search time, making it highly efficient in terms of computational speed.
    c. **Pros: Fast search time**
    d. **Cons: Low precision and recall**

3. **Autoencoder -**
    a. Highest search time, which may hinder real-time applicability and scalability.
    b. Good for applications where accuracy is more critical than speed.
    c. **Pros: High accuracy metrics**
    d. **Cons: High search time**

4. **Custom CNN –**
    a. Outperforms all other methods in Precision@K, Recall@K and matches the highest retrieval accuracy.
    b. Search time is relatively low.
    c. Provides a good balance between accuracy and computational efficiency which ensure scalability.
    d. **Pros: High precision and recall, low search time.**
    e. **Cons: Slightly higher computational cost.**

5. **Variational Autoencoder –**
    a. Performs similar to the Autoencoder except it has a lower search time.
    b. More suitable for practical applications as compared to Autoencoder.
    c. **Pros: High accuracy with moderate search time.**
    d. **Cons: Complex to implement.**

Performance v/s Computational Efficiency

- **Best Overall Performance: Custom CNN**
    - Highest Precision and Recall
    - Matches the highest retrieval accuracy.
    - Low search time.

Use Cases Approach:

- **Large-Scale Systems with Limited Computational Resources:**
  Hashing Based Method since it provides the necessary speed for handling large datasets despite low accuracy, other models who offer a low search time with a moderate accuracy can be considered.

- **Real-Time Systems with High Accuracy Needs:**
  Custom CNN since it provides a superior accuracy along with a reasonable search time speed as compare to other models.

- **Complex System**
  Custom CNN and Variational Autoencoder are two models that captures more complex features and based on accuracy or search time can be selected to serve a task.

Handling Various Image Types:

- **Complex or High-Variability Datasets:**
  - Custom CNN and Variational Autoencoder can be considered since they capture more complex features

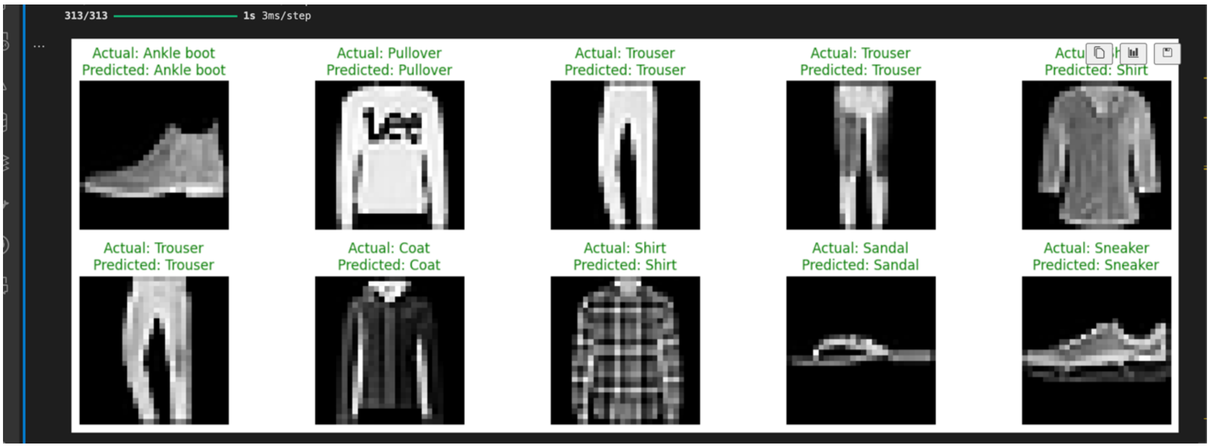
- **Simpler or Uniform Datasets:**
  - Siamese Network and Hashing Based methods can be used because of their low computational requirements.

Scalability Considerations:

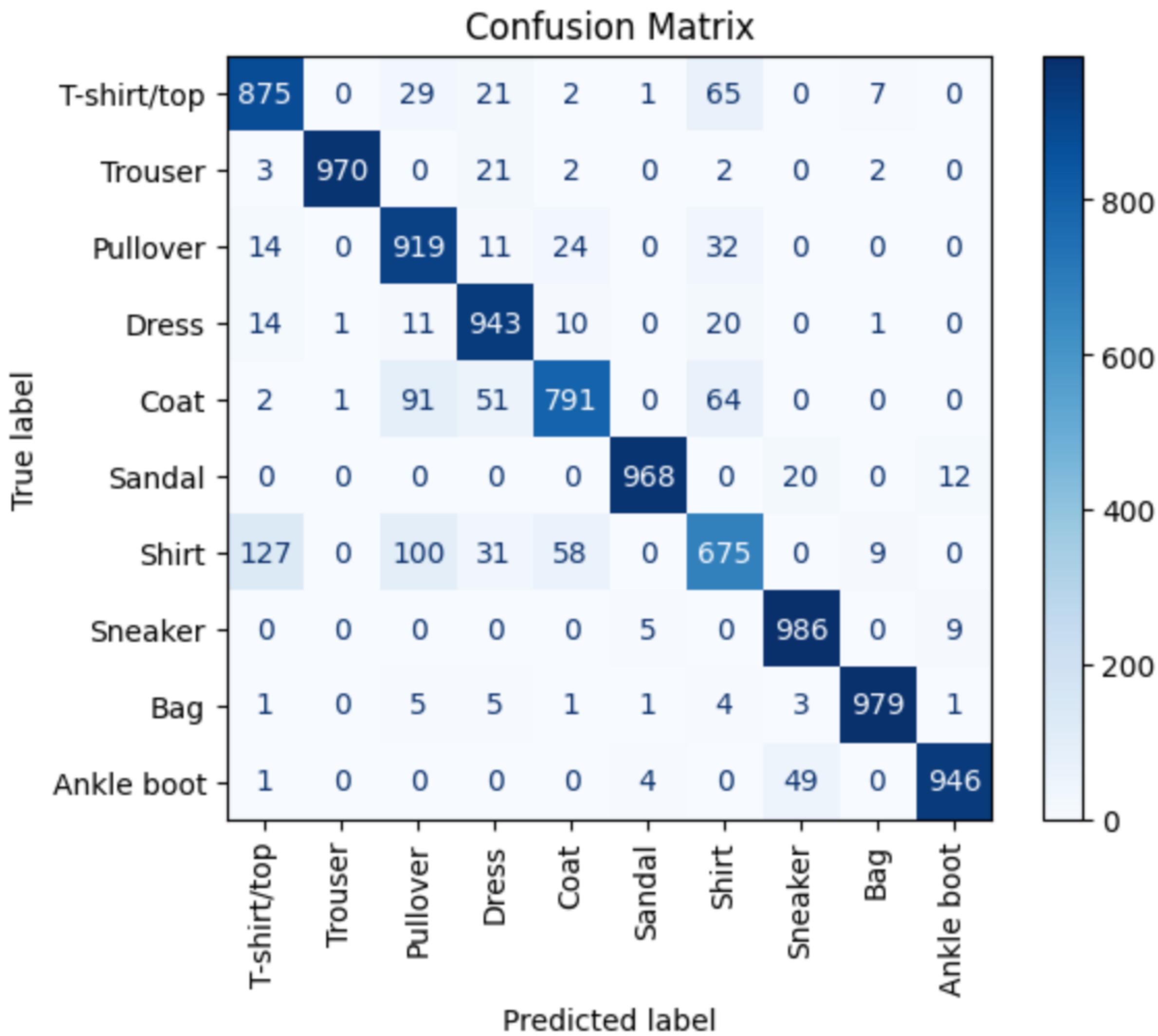
- **Based on Memory Usage**
  - Custom CNN and Autoencoder models require more memory to store high-dimensional embedding leading to optimization for larger datasets.
  - Hashing Based models have compact embeddings leading to lower memory, and is more easily scalable even with a large dataset.

- **Based on Optimization**
  - Custom CNN and Variational Autoencoder can leverage batch processing and GPU acceleration during inference.
  - Other models can be implemented for high-dimensional embedding to enhance scalability and speed up similarity search.

Results of Custom CNN prediction (Best model out of the 5)

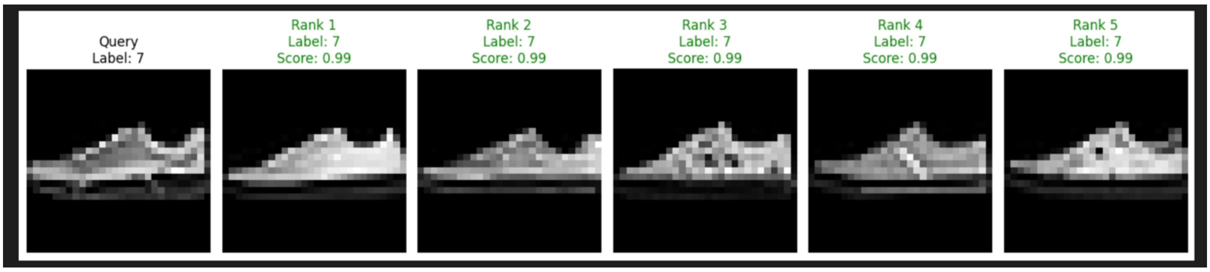


Below is a Confusion Matrix, and we can see the number of correct and wrong predictions by the Custom CNN model.

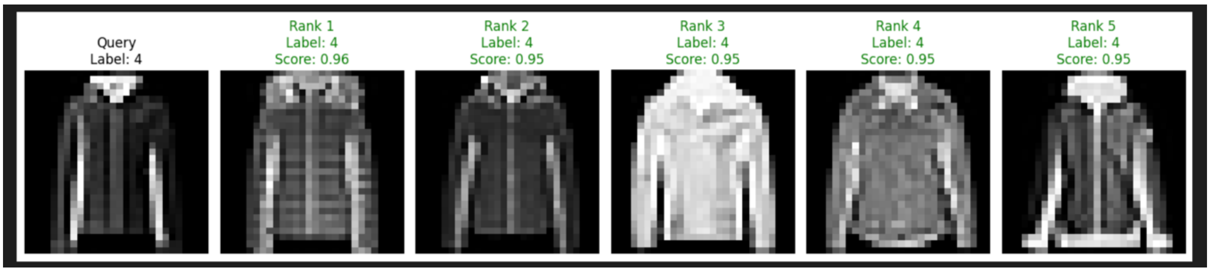


Similarity Seach for different labels:

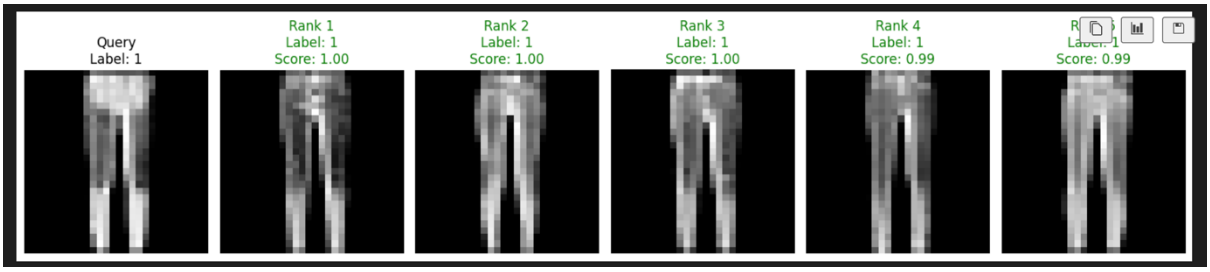
Sneakers-



Dress-



Trousers-



Shirt-