AI-Powered Image Similarity Search and Recommendation System

This project report is submitted under
Intel® Unnati Industrial Training Program 2025

for successful completion of the

Of

Intel® Unnati Industrial Training Program 2025

Submitted By

Debasrita Chattopadhyay

Krutika Kailash Umale

B.Tech – Artificial Intelligence and Data Science
Yeshwantrao Chavan College of Engineering
Nagpur, Maharashtra, India

Under the guidance of

Supriya Thombre

Table of Content

ABSTRACT

With the rapid increase in image data over digital platforms, traditional text-based image retrieval methods have become ineffective due to not being able to identify and understand visual traits like texture, colour and shape. Therefore, this project proposes the development of a deep-learning image comparison search system which does not require the use of any text-based metadata. Instead, this system will use a form of deep learning called triplet networks with triplet loss to create a discriminative embedding space where visually similar images are located closer in proximity to each other. The encoder within the new search system will generate compact, 128-dimensional representations (embeddings) of all images. Two images can be compared quickly and efficiently using cosine similarity on their respective embeddings. The developed model will be implemented within a Flask web application to provide users with a simple way to upload their chosen image and return a list of similar looking images that are visually appealing.

1.0 INTRODUCTION

1.1 Overview

When the massive amounts of digital images are being uploaded in various domains, such as e-commerce, fashion and social media, coming up with an efficient way to find similar images has become a pressing need. Traditional text-based search is no longer able to keep up with the task, manual tagging is laborious, and misses the mark on visual similarity.

This project presents a brand-new AI-powered image similarity search and recommendation system that uses deep learning to find visually similar images, by breaking down the process into smaller components. The system's triplet network with MobileNetV2 is able to dig out meaningful patterns in the images and put similar-looking images close to each other in the "embedding space".

The trained model indexes a massive image database and can perform lightning-fast similarity searches. It does this by applying the cosine similarity algorithm and also includes a user-friendly web-based interface that lets users upload and receive real-time results, turning the system into

something that's practical, expansive and works well in the real world, and will be super useful for product recommendations and visual search.

1.2 Problem Statement

Although there has been great advancement in the area of Image Classification, the majority of the current systems do not have the means to optimize for Similarity-based Retrieval. Classification models focus on assigning an image to just one label, whereas Similarity Search focuses on understanding how the images in your dataset relate to one another in terms of what makes them visually similar. In certain fields, such as Fashion Recommendation Systems, two visually similar products may belong to different Categories or Subcategories making it difficult for Classification Models to be used effectively.

Due to the large number of images in most datasets, image storage requirements and speed of retrieval become very important. Moreover, having a large number of features in the high-dimensional feature vectors makes them take longer to compute and more memory-intensive, making them unusable for real-time operations. As a result, the major problem being solved by this project is the Development of a System that:

- Learns and stores compact, yet discriminative, visual representations
- Retrieves visually similar images with accuracy
- Scales to accommodate increasing amounts of Data
- Delivers real-time responses for user-facing applications

1.3 Objectives

The goals of this project are focused on three main areas.

- The creation of a computer vision infrastructure which will make it possible to perform image similarity searches. This will be done through content-based analysis (not relying on text labels), thus enabling users to find visually similar images to one another.
- Utilization of a Triplet Network structure to create image embeddings which enhance the ability of the system to differentiate between similar and dissimilar images by reducing the distance between similar images and increasing the distance between dissimilar images.

- To demonstrate how image similarity searches can be useful in many business settings including fashion and e-commerce product searches, visual content discovery platforms, and as a method of improving the experience of a customer when they are looking to make a purchase.

2.0 LITERATURE REVIEW

For decades, computer vision researchers have been investigating image similarity search as a research discipline. The majority of early image retrieval systems were based on manually designed visual descriptors (e.g., Scale-Invariant Feature Transform (SIFT) by Lowe (2004), and Histogram of Oriented Gradients (HOG) by Dalal and Triggs (2005)). These types of systems were limited to low-level features, such as edges, gradients and key points. Although effective in laboratory settings, these two-dimensional features did not work well in the real world because they could not handle variations due to changes in lighting, occlusions, or excess background clutter. Moreover, these systems could not incorporate high-level semantic information from the images that could aid search results.

The rise of deep learning has transformed how image data is represented using CNNs (Krizhevsky et al. (2012)). AlexNet demonstrated that deep CNN architectures could learn representations of the visual hierarchy through the use of large datasets; beating state-of-the-art performances in comparison to traditional handcrafted feature descriptors. Some of the best CNN architectures to date include VGGNet (Simonyan and Zisserman (2015) and ResNet (He et al. (2016), which improved on prior work by deepening the model and including residual connections between the layers. Even though many of these architectures were trained for image classification tasks, they were not designed specifically for image retrieval.

To address these limitations with classification-based data presentations, there is an increasing focus on metric learning using techniques such as the Siamese network developed by Bromley et al. (1993); which creates similarity measures by minimizing distance metrics between pairs of images.

3.0 Dataset Description

The Clothing Dataset (small) was used within this project. The dataset is a publicly available dataset and contains several different categories of clothing including: shirts, dresses, trousers, shoes, and accessories. The dataset is organized as a directory structure that is class-based, making it compatible with triplet-based learning. Dataset Description is illustrated in Table 3.1.

Table 3.1 Dataset Description

Attribute	Description
Dataset Name	Clothing Dataset Small
Image Format	JPEG
Image Size	128 × 128
Categories	Multiple clothing types
Total Images Used	3068

4.0 System Architecture

There are three main parts to the proposed system architecture: the training, indexing and inference components (as illustrated in Fig 4.1). The heart of the system is made up of the Triplet Network (as shown in Fig 4.2) which includes three identical CNN encoders that share weights. The encoders take in one of the three images (the triplet) as input, namely the anchor image, consecutive positive and negative triplet images.

The encoder has been designed based off of MobileNetV2, which is an extremely lightweight convolutional neural network that has already been pre-trained on ImageNet. MobileNetV2 has been selected because of its speed and efficiency, smaller parameter size and ability to be deployed in resource-limited environments. The encoder's output is processed via a global-average pooling layer followed by a fully-connected layer, resulting in a 128-dimensional embedding that is L2-normalized to enable more reliable similarities to be computed between embeddings.

Only the end-user encoder will generate embeddings from query images when making use of the inference component. The use of one encoder results in the generation of embeddings quickly with a minimum memory requirement.

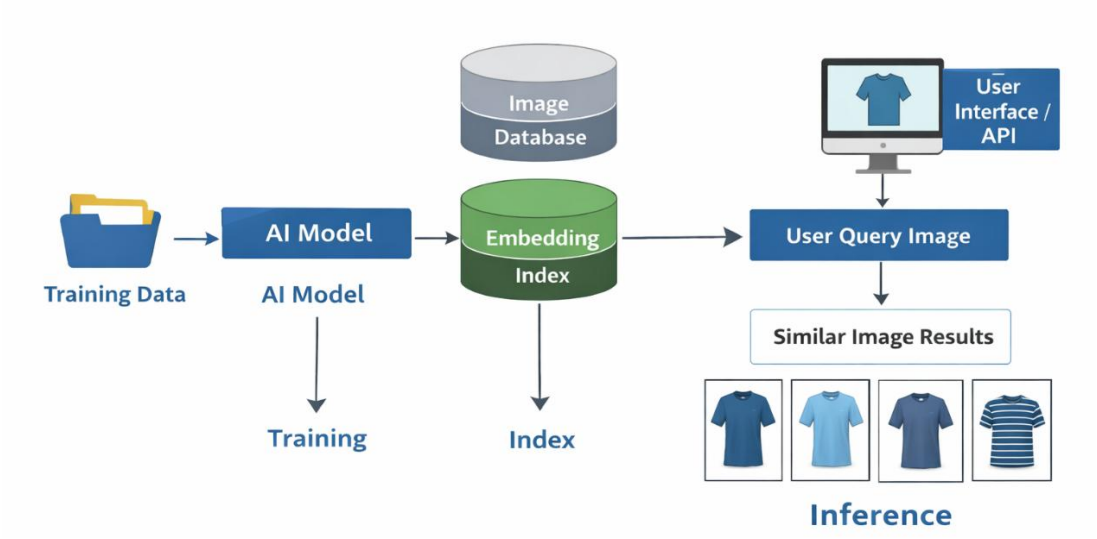

Fig 4.1 System Architecture diagram

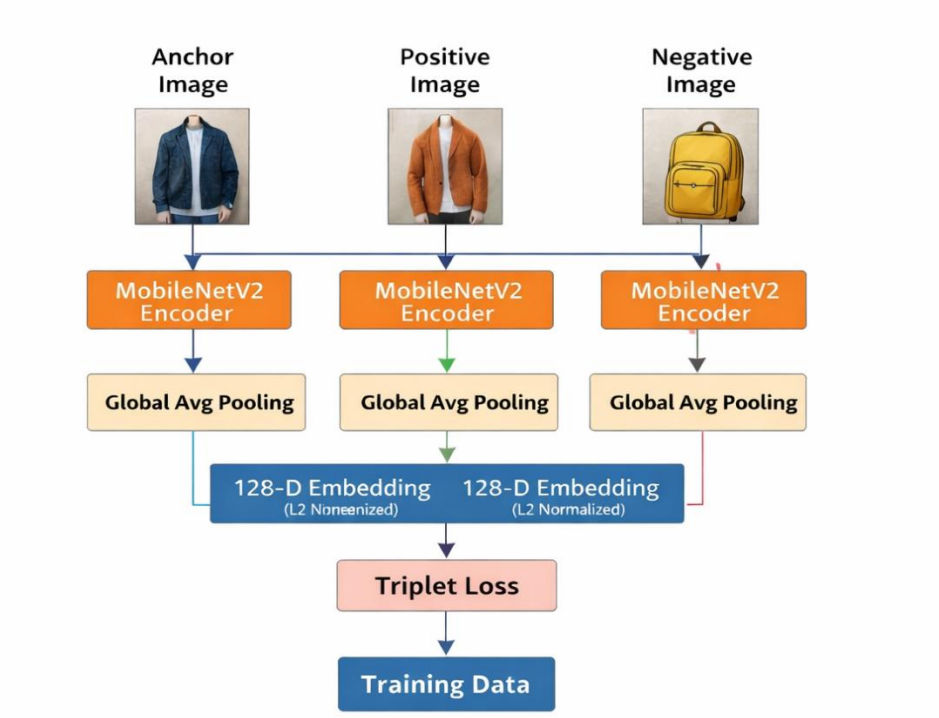

Fig 4.1 Triplet Network Architecture Diagram

5.0 Solution Implementation

Training generates Image Triplets in real time. Each triplet contains:

- An anchor image

- A positive image from the same category
- A negative image from a different category

The Triplet Loss Function guarantees that the distance between the anchor and positive embeddings will always be smaller than the distance between the anchor and negative embeddings by a fixed margin. Embeddings are created and saved for the entire database of images after the training has been completed. Once indexing has been done on these embeddings, they can efficiently perform Similarity Search through Cosine Similarity when inferring.

6.0 Results and Analysis

6.1 Training Output

Training the model consisted of 5 Epochs (steps = 100 for each epoch). Each triplet loss reduced consistently (Table 6.1) through training indicating that the model was learning how to create visual embeddings that minimize distance between each image's visual representation relative to the visual representation of other similar images.

Table 6.1 Training Loss Across Epochs

Epoch	Steps per Epoch	Triplet Loss
1	100	0.1495
2	100	0.0676
3	100	0.0615
4	100	0.0449
5	100	0.0377

Interpretation:
The consistent decrease in triplet loss is evidence of a good metric learning capability. The high rate of decline noted at the beginning of the training process is attributed to the use of transfer learning techniques, while there's a slower but consistent improvement of the embedded feature vectors produced in the later epochs.

6.2 Database Indexing Results

Once the entire images dataset has been trained, the dataset can be indexed using the embeddings created from all of the images. Database Indexing Summary is illustrated in Table 6.2.

Table 6.2 Database Indexing Summary

Parameter	Value
Total Images Indexed	3068
Embedding Dimension	128
Database Shape	(3068, 128)

Interpretation:
Each image is represented by a compact feature vector, enabling fast similarity search and reduced storage overhead.

6.3 Embedding Space Characteristics

A summary of Embedding Space Properties is demonstrated in Table 6.3.

Table 6.3 Embedding Space Properties

Feature	Observation
Intra-class similarity	High
Inter-class separation	Clear
Normalization	L2
Similarity Metric	Cosine similarity

Clusters of visually similar items like clothing items exist within the geometric space of the embedding vectors which confirm where similar styles of clothing will typically exist.

6.4 Visual Output Results

The results of the Fashion Lens System are presented in Section 6. The example images illustrate how the Fashion Lens System can use deep learning-based similarity matching capabilities to retrieve and recommend visually similar fashion products to a user based solely on image content. The results demonstrate that the Triplet Network is effective in learning visual semantics and delivering accurate, category consistent, and real-time recommendation results.

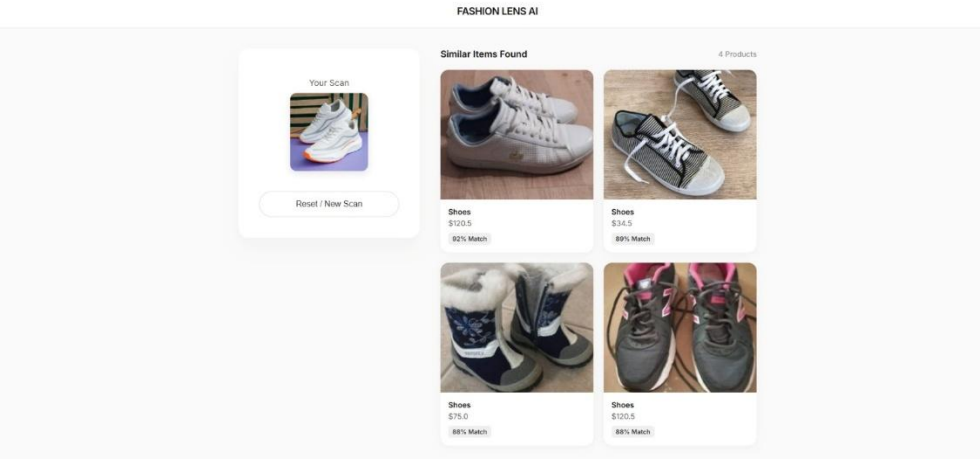

Fig 6.4 Image similarity search results for a footwear query showing visually similar shoe products retrieved by Fashion Lens.

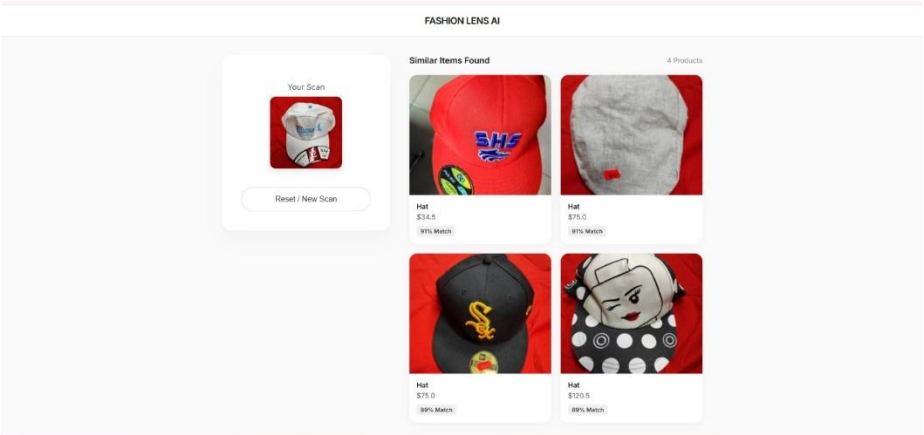

Fig 6.5 Similar item recommendations for a headwear (cap) query, demonstrating category-specific visual matching.

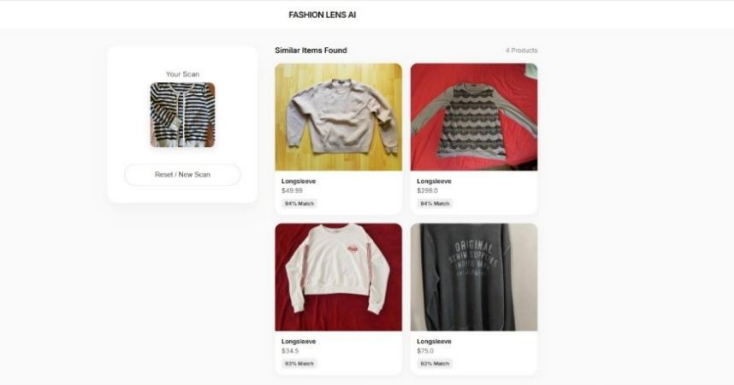

Fig 6.5 Long-sleeve apparel similarity results highlighting pattern, color, and style-based matching.

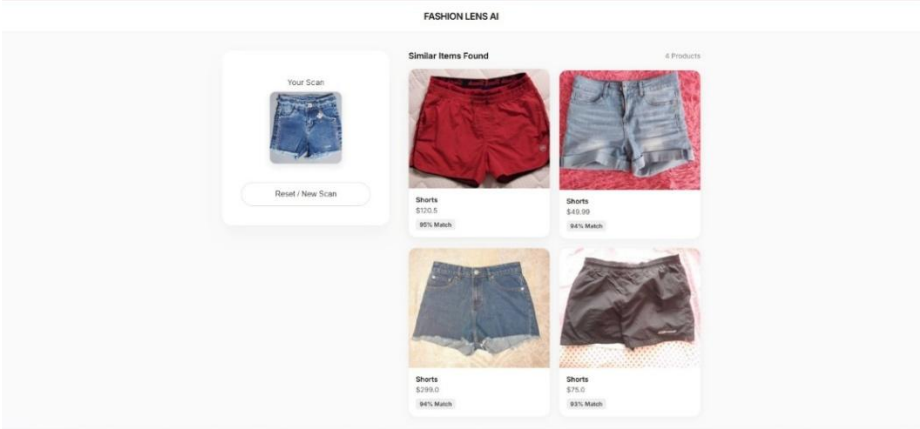

Fig 6.6 Shorts product similarity results showcasing accurate retrieval based on garment type and visual features.

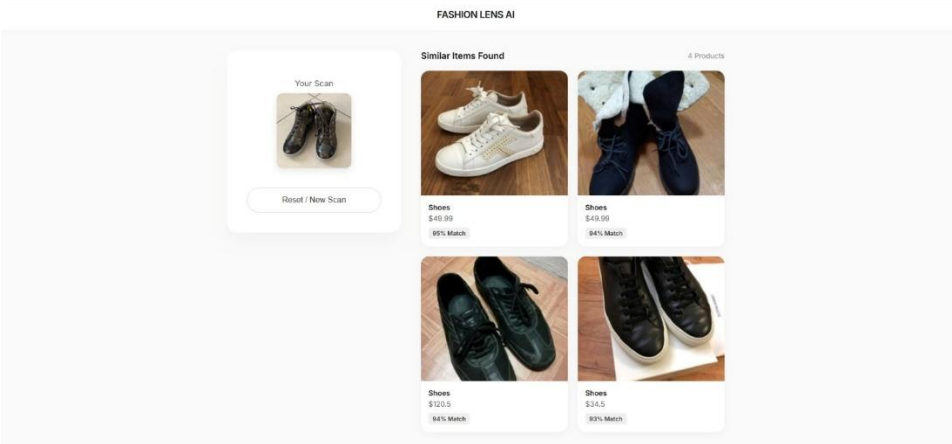

Fig 6.7 Visual similarity search results for a shoe query image retrieved using the Fashion Lens system.

7.0 Performance Evaluation

The system provides near real-time query results and, for the indexed dataset, provides retrieval of similar images within an average of 1 second. Cosine Similarity and the use of compact embedded vectors yield fast computation for the distance metric. The MobileNetV2 backbone architecture decreases the time it takes to perform inference, but does not degrade the quality of the retrieval results. The System Performance Metrics are included in Table 7.1.

Table 7.1 System Performance Metrics

Metric	Observation
Query Response Time	< 1 second
Scalability	3000+ images
Model Backbone	MobileNetV2
Retrieval Method	Embedding similarity

The system meets real-time performance requirements and scales effectively for medium-sized datasets.

Epoch-loss visualization

The Epoch Loss Graph (Fig.7.1) shows how the training of the Triplet Network progresses through an Epoch. The continued loss value reduction as you complete more Epochs indicates the model is learning and getting closer to the Anchor-Positive image while becoming further away from the Anchor-Negative image. As the loss of the model stabilizes later during the Epoch, this helps to ensure that overfitting is reduced and that the model has converged.

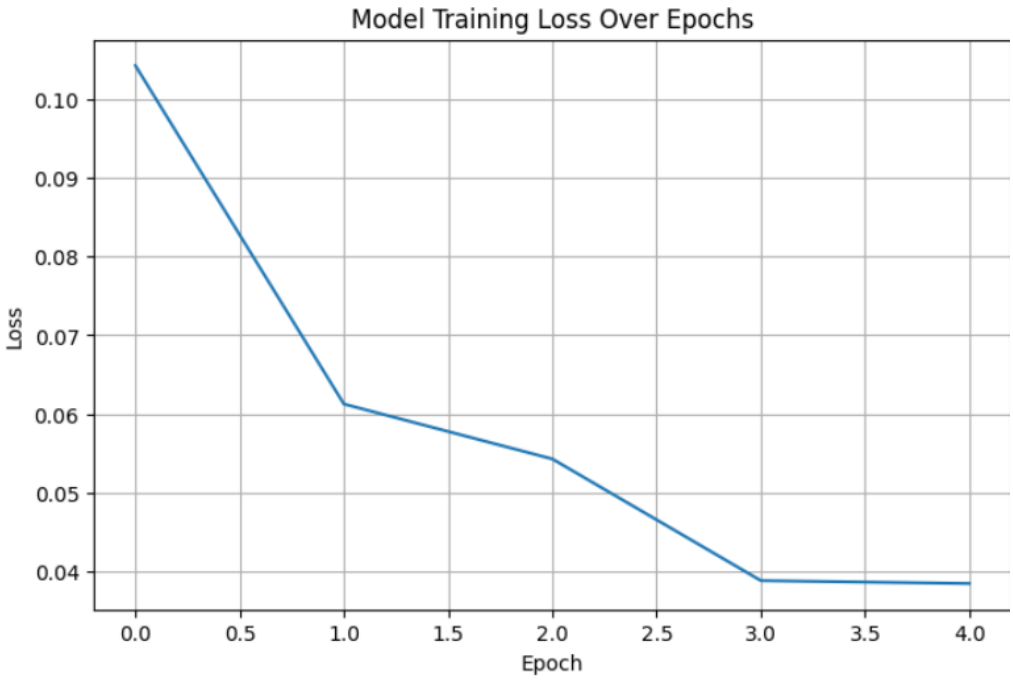

Fig 7.1 Epoch Loss Graph

7.3 Triplet Evaluation

The Triplet performance is determined by whether Anchor–Positive Distance < Anchor–Negative Distance. .The model achieved a success rate of 97.50%, which means it is able to learn and associate visual similarity with a very high success rate. This indicates the structure of the model-generated embeddings are well-defined and will produce the most relevant results for visual similarity search and recommendation.

8.0 Conclusion

This project has demonstrated that an AI-based image similarity search/results recommendation system can be developed and built using Deep Metric Learning techniques. The project utilized Triplet Networks and MobileNetV2 to produce images with a high degree of accuracy, efficiency and scalability. The results indicate that the system works as intended and is suitable for deployment in a live environment. The findings of this project show the opportunities for developing deep learning-based image similarity search systems to address the shortcomings of traditional methods.

9.0 Future Scope

Future improvements that may be possible include:

- Continued tuning of the encoder on domain-specific datasets to further define accuracy.
- Utilization of large-scale indexing frameworks, such as FAISS, to support scale.
- Development of a multi-modal search capability for image and text input to enhance user interaction.
- Provide access to the system through a mobile or cloud-based application for greater usability.

10.0 References

[1]. Lowe, D. G. (2004). *Distinctive image features from scale-invariant keypoints*. International Journal of Computer Vision, 60(2), 91–110.

[2]. Dalal, N., & Triggs, B. (2005). *Histograms of oriented gradients for human detection*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[3]. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. Advances in Neural Information Processing Systems (NeurIPS).

[4]. Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. International Conference on Learning Representations (ICLR).

[5]. He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition*. IEEE CVPR.

[6]. Bromley, J., et al. (1993). *Signature verification using a "Siamese" time delay neural network*. Advances in Neural Information Processing Systems.

[7]. Howard, A. G., et al. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications*. arXiv preprint arXiv:1704.04861.

[8]. Sandler, M., et al. (2018). *MobileNetV2: Inverted residuals and linear bottlenecks*. IEEE CVPR.

[9]. Wang, J., et al. (2017). *Deep metric learning with cosine similarity*. IEEE ICCV Workshops.