HW to Chapter 18 “Neural Style Transfer”

***Non-programming Assignment***

1. What is face verification and how does it work?
2. Describe the difference between face verification and face recognition?
3. How do you measure similarity of images?
4. Describe Siamese networks.
5. What is triplet loss and why is it needed?
6. What is neural style transfer (NST) and how does it work?
7. Describe style cost function.

Answer

1. **What is face verification and how does it work?**

Face verification is a type of biometric authentication that confirms whether an inputted image matches the same image being compared. Unlike bigger recognition systems, facial verification focuses on binary validation which is a one-to-one comparison. It works by: Taking an input image along with a claimed name or ID, using a model trained to compare the facial features in the input image with a stored reference image, outputting a result indicating whether the input image is of the claimed person.

The workflow involves:

* Feature Extraction: A deep neural network (typically a pre-trained convolutional neural network) transforms input face images into compact, meaningful vector representations called embeddings. These embeddings capture intrinsic facial characteristics while being invariant to minor variations in pose, lighting, and expression.
* Similarity Computation: The extracted embeddings are compared using a distance metric, like Euclidean distance.
* Decision Threshold: A predefined similarity threshold determines whether the two faces are considered a match. This threshold is calibrated during model training to balance false acceptance and false rejection rates.

The process essentially transforms the verification problem from a pixel-level comparison to a learned, high-dimensional feature space where similar faces cluster together.

1. **Describe the difference between face verification and face recognition.**

* **Face Verification**: This is a **one-to-one** matching process where the system checks if an input image matches a specific claimed identity. It is a binary decision (same/different person) typically used for authentication scenarios and has lower computational complexity. It is commonly used in authentication systems like unlocking a phone with Face ID and access control systems.
* **Face Recognition**: This is a **one-to-many** identification process where the system compares an input image against a database of multiple faces and identifies the person or returns “not recognized” if the face is not found in the database. It does have a higher computational complexity. Some example use cases are law enforcement, large-scale surveillance systems, etc.

The key distinction lies in the comparison scope: verification validates identity against a claimed identity, while recognition attempts to determine identity from scratch.

1. **How do you measure the similarity of images?**

The similarity of images is measured using distance metrics, which quantify how close or far two image feature representations are. Common metrics include Euclidean Distance which measures the straight-line distance between two feature vectors. Cosine Similarity measures the cosine of the angle between two feature vectors. Triplet Loss ensures that similar images are closer in the learned embedding space, while dissimilar ones are farther apart.

In detail,

**Feature-Based Metrics**:

* Cosine Similarity: Measures angle between feature vectors
* Euclidean Distance: Calculates geometric distance between embeddings
* Manhattan Distance: Computes sum of absolute differences

**Deep Learning Approaches**:

* Learned Metric Learning: Neural networks explicitly trained to generate similarity-preserving embeddings
* Siamese Networks: Architectures designed to learn comparative representations
* Triplet Networks: Learn discriminative embeddings by simultaneously considering anchor, positive, and negative samples

1. **Describe Siamese Networks.**

A Siamese Network (also called a twin neural network) is a type of artificial neural network that uses the same weights while processing two different input images. The network produces comparable output vectors for the images and measures the similarity of the two outputs using a distance function.

**Architectural Characteristics**:

* Twin neural network branches with identical architectures
* Shared weights ensure consistent feature extraction
* Final layer typically produces a similarity score or distance metric

**Training Dynamics**:

* Networks learn to map similar inputs closer together in embedding space
* Dissimilar inputs are pushed further apart
* Loss functions like contrastive or triplet loss guide this learning process

**Key Advantages**:

* Can learn similarity metrics from limited training data
* Inherently robust to variations within a class
* Enables few-shot and zero-shot learning scenarios

**Applications of Siamese Networks include:**

* Face verification: Comparing a given face to a reference image.
* Handwritten signature verification.
* Matching queries with indexed documents.

1. **What is triplet loss and why is it needed?**

Triplet loss is a distance-based loss function used to train models to differentiate between similar and dissimilar images. It works by using triplets of images: an anchor (A), a positive (P) (same class as the anchor), and a negative (N) (different class). Ensuring that the distance between the anchor and positive is smaller than the distance between the anchor and negative by a margin ε.

**The Mathematical formular:**

L = max (|| f(anchor) - f(positive) || - || f(anchor) - f(negative) || + margin, 0)

Triplet loss helps in face recognition by learning a feature space where faces of the same person are closer together, while faces of different people are farther apart.   
   
**Purpose**:

* Enforces embedding space constraints
* Ensures similar samples cluster together
* Pushes dissimilar samples apart
* Provides more nuanced learning compared to pairwise approaches

1. **What is Neural Style Transfer (NST) and how does it work?**

Neural Style Transfer (NST) is a deep learning technique that applies the visual style of one image to the content of another. It works by taking a content image (C) and a style image (S). Using a deep convolutional neural network (CNN) (e.g., VGG-19) to extract content features from a deeper layer and style features from shallower layers. Generating a new image G, which is optimized to minimize:

* Content loss: Difference between the content of G and C.
* Style loss: Difference between the style of G and S using a Gram matrix.

Iteratively updating G using gradient descent to create the final stylized image.

1. **Describe the style cost function.**

The style cost function quantifies the difference in style between the generated image and the style image. It is calculated using a Gram matrix, which captures the correlations between feature maps at different layers.

* The cost function is:

**Lstyle = ∑ wl \* 1/ [4N^2C^2] || G^l − A^l ||^2**

where:

* G^l is the Gram matrix of the generated image at layer ll.
* A^l is the Gram matrix of the style image at layer ll.
* wl is a weighting factor for each layer.

This function ensures that the texture and style patterns from the style image are applied to the generated image.