Chapter 3: Attention and Embedding Models in Biometric Recognition

Attention-based deep networks have recently revolutionized biometric feature learning. Instead of relying solely on local convolutional filters, self-attention mechanisms allow a model to consider all parts of an input simultaneously, weighing their importance when computing a representation. In a Transformer's self-attention block, each token (e.g., an image patch or time step) produces a query vector, and all tokens produce key and value vectors. The query of one token is compared (via dot-product) with the keys of all tokens to compute attention weights, and the weighted sum of values gives the output for that token. In effect, every patch "looks at" every other patch in the image (or every frame in a sequence), capturing long-range dependencies that CNNs with limited receptive fields may miss. Multi-head attention extends this by computing several sets of queries, keys, and values in parallel, so that different "heads" can focus on different patterns (e.g., one head might focus on eyes, another on mouth in a face image) [r]. Crucially, Transformers use Layer Normalization (instead of batch norm) to stabilize training: each token's features are normalized across the embedding dimension after each attention and feedforward block. This ensures that the scale of the embeddings remains consistent across layers, which empirically helps convergence in large models.

Self-attention is naturally suited to modeling biometric data. For example, in a fingerprint image, a distant minutia point can influence the interpretation of another (ensuring global ridge continuity), while in face images, each patch can attend to all facial regions (improving occlusion robustness) [r]. Transformers have been applied across modalities: beyond faces and fingerprints, they have been used for finger-vein, iris, gait, and even keystroke dynamics. For instance, a recent "AuthFormer" model for multimodal biometrics uses a Vision Transformer for image modalities (face, finger) and a Transformer for sequential signals (keystrokes) to jointly model different traits [r]. In that work, raw biometric sequences are first embedded (e.g., face patches encoded by ViT embeddings; keypress sequences encoded by learned positional embeddings) and then fused via self-attention. The ability of self-attention to integrate heterogeneous inputs makes such multimodal models highly flexible.

Embedding Representations for Biometric Identity

Modern biometric recognition systems typically map each sample (e.g., an image of a face or fingerprint) to a fixed-length embedding vector in high-dimensional space. This embedding is trained such that samples of the *same* person are close together and samples of *different* persons are far apart. In practice, a deep network (CNN or Transformer) is trained on a large dataset of identities so that its penultimate layer outputs are identity embeddings. For face recognition, classic examples include FaceNet and ArcFace, which learn 128–512 dimensional embeddings via triplet loss or margin-based softmax losses [r]. Similarly, fingerprint recognition can use CNN or attention models to extract a global fingerprint vector; for instance, *AFR-Net* combines ViTs and ResNets to produce a global fingerprint embedding that outperformed commercial systems [r].

Key practical points in embedding design include:

- Loss functions: Beyond simple softmax classification, many systems use metric losses (triplet, contrastive) or angular-margin losses (ArcFace, CosFace) to directly enforce angular separation in embedding space. These losses often normalize embeddings to lie on a hypersphere (unit L2 norm) before computing similarities. For example, CosFace explicitly L2-normalizes both feature vectors and weight vectors so that the decision boundary depends only on the cosine of the angle between them. This places all embeddings on a common scale, which has been shown to improve discrimination (maximizing inter-class angles while minimizing intra-class variance).

- Embedding dimension: Typical face embeddings range from 128–512 dimensions, while fingerprint embeddings can be similar. Higher dimensions can encode more detail but require more storage and computation. In practice, these dimensions are chosen to balance accuracy and efficiency.

- Canonical embeddings: Many works pre-train a CNN on face classification (thousands of identities) and then use the activations as embeddings. In Transformer-based biometrics, a common approach is transfer learning: a ViT pre-trained on ImageNet is fine-tuned on the biometric dataset to produce embeddings. For example, vision transformers have been fine-tuned for finger-vein and vein recognition, showing strong performance even with limited data [r].

Typical Biometric Embedding Pipelines

1. Feature Extraction: Input (e.g., face image) ➔ CNN/ViT ➔ fixed-length vector (embedding).

2. Normalization: Often, embeddings are L2-normalized. Some systems also apply batch or layer normalization within the network to stabilize features.

3. Similarity Matching: Cosine or Euclidean distance between embeddings. A decision threshold on this distance determines a match.

4. Score Normalization (post-hoc): In biometrics, score normalization (e.g., Z-norm, T-norm) can adjust similarity scores to improve comparability. Recent work shows that cohort-based score normalization can mitigate demographic biases by equalizing score distributions across groups [r].

These embeddings have broad applications: in verification (1:1 match) or identification (1:N search), and even for forming *cancelable templates* (see Chapter 4). Empirically, transformer-based embeddings have proven effective. For instance, the **LVFace** model (a large ViT trained on 42M faces) achieved state-of-the-art face recognition accuracy on challenging benchmarks by leveraging transformer embeddings and modern losses [r]. Similarly, in fingerprints, attention-based AFR-Net combined ViT and CNN embeddings to surpass prior CNN-only systems.

Normalization and Loss Techniques

Normalization in the context of embedding learning comes in several forms. Inside the network, *Layer Normalization* (as used in Transformers) scales each token's features to zero mean and unit variance across the feature dimensions. This contrasts with *Batch Normalization* (common in CNNs) which normalizes across the batch dimension; layer norm is essential in attention models to ensure stable gradients regardless of batch size.

For the embeddings themselves, it has become standard to apply an L2 normalization before computing loss. For example, CosFace reformulated softmax into an “angular” softmax by normalizing both the feature and weight vectors to length 1. This means all identity vectors lie on a hypersphere, and classification depends only on the angle (cosine similarity) between vectors. ArcFace similarly normalizes embeddings and adds an angular margin to further separate classes. These normalization steps remove radial (magnitude) variance and focus learning on angular discrimination.

After training, score (distance) normalization may also be applied. Unlike feature normalization, score normalization (e.g. Z-norm) adjusts raw similarity scores based on cohort statistics. Recent research shows cohort-based normalization can improve fairness: by normalizing impostor and genuine score distributions within each demographic group, overall False Match/Non-Match rates become more uniform across groups. Such post-processing is complementary to network design and can be applied to any pre-trained model to reduce demographic bias without retraining.

Finally, specialized loss functions are often used to improve embedding quality. Triplet and contrastive losses explicitly shape the embedding space by pulling same-class samples together and pushing different-class samples apart. Margin-based softmax losses (ArcFace, CosFace, SphereFace) incorporate geometric margins in angular space. As cited earlier, CosFace (Large Margin Cosine Loss) shows analytically why L2-normalizing features leads to a uniform angular spread, aiding discrimination. In practice, combining these losses with attention models has led to significant gains. For example, LVFace combined transformer training with state-of-the-art face losses (ArcFace, CosFace, etc.) to achieve 1st place on a major challenge.

Attention in Practice: Examples

- Fingerprint Recognition: AFR-Net uses a shared CNN+Transformer backbone to extract fingerprint embeddings. It further refines global embeddings by aligning local patch correspondences between fingerprint pairs. This means that if two fingerprint images partially overlap, the model can focus its attention on matching ridge regions, boosting accuracy in hard cases.

- Ear and Vein Biometrics: Surveys report transformers being fine-tuned for unconstrained ear recognition and various vein modalities. For instance, Garcia-Martin et al. applied several pre-trained ViTs to vascular biometrics (finger, palm, wrist veins) and showed near-perfect identification rates on public datasets. The patch-based ViT could leverage the fine texture patterns in veins across the hand, emphasizing the global context of vein patterns.

- Keystroke Dynamics: In free-text keystroke authentication, a Transformer model learned inter-key timing dependencies. A comparative study found that a Transformer with batch-all triplet loss achieved an EER of 0.0186%, outperforming RNNs [r]. The self-attention mechanism was able to capture long-range timing patterns (like rhythms and habits) that characterize an individual’s typing style. This illustrates that attention can be powerful even for sequential, behavioral biometric data.