ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

CSC14006 - Nhận dạng

Báo cáo Seminar

Đề tài:

Facial Landmark Localization

Sinh viên thực hiện:

Nhóm 10

Nguyễn Văn Minh Thiện

Vũ Minh Phát

Nguyễn Quang Thông

Nguyễn Trọng Tín

Giảng viên hướng dẫn:

PGS. TS. Lê Hoàng Thái

ThS. Dương Thái Bảo

ThS. Trương Tấn Khoa

Ngày 12 tháng 5 năm 2025

Thông tin nhóm

Lớp: Nhận dạng - 22KHMT

Sinh viên thực hiện: Nhóm 10

STT	MSSV	Họ và tên	Mức độ đóng góp
1	21127731	Nguyễn Trọng Tín	100%
2	21127739	Vũ Minh Phát	100%
3	22127398	Nguyễn Văn Minh Thiện	100%
4	22127401	Nguyễn Quang Thông	100%

Mục lục

1 Chương 1: Giới thiệu

Tên chủ đề: Facial Landmark Localization (*tạm dịch*: định vị điểm mốc trên khuôn mặt)

Từ khóa: Facial Landmark Detection, Facial Alignment, ...

1.1 Tổng quan về chủ đề

Nhận diện khuôn mặt đã nổi lên như một lĩnh vực sôi động trong sinh trắc học, thu hút sự quan tâm nghiên cứu và ứng dụng thương mại mạnh mẽ trong suốt nhiều thập kỷ qua. Nhiệm vụ chính của nhận diện khuôn mặt là phát hiện sự hiện diện của khuôn mặt trong ảnh và xác định vị trí của chúng thông qua khung giới hạn (bounding box). Các nghiên cứu gần đây đã chứng minh rằng công nghệ nhận diện khuôn mặt đã đạt đến trình độ tiên tiến cả về độ chính xác lẫn tốc độ xử lý [1].

Tuy nhiên, chỉ phát hiện khuôn mặt thôi là chưa đủ để thu thập được các điểm mốc (hay còn gọi là điểm đặc trưng) trên khuôn mặt, ví dụ như đường viền mắt, khóe miệng, mũi, lông mày, v.v.. Đây chính là mục tiêu của **bài toán định vị điểm mốc trên khuôn mặt (facial landmark localization)**, một tác vụ nhằm xác định chính xác vị trí của các điểm đặc trưng trên khuôn mặt (như được minh họa trong Hình 1).

1.2 Động lực nghiên cứu

Facial Landmark Localization là bài toán quan trọng về cả mặt khoa học và ứng dụng. Về khoa học, đây là nền tảng để giải quyết các vấn đề tối ưu vị trí điểm trong không gian ảnh, phát triển các mô hình học máy tiên tiến, và nghiên cứu lý thuyết biểu diễn đặc trưng trong thị giác máy tính. Về ứng dụng, bài toán này phục vụ thiết thực cho nhận dạng khuôn mặt, phân tích cảm xúc, an ninh, giải trí, tương tác người-máy và nhiều lĩnh vực công nghệ khác.

1.3 Tính ứng dụng của chủ đề

Việc xác định chính xác các điểm đặc trưng trên khuôn mặt là yếu tố then chốt trong nhiều ứng dụng thực tế, bao gồm nhưng không giới hạn:

- ***Nhận diện khuôn mặt*** *(Face Recognition):* Landmark hỗ trợ căn chỉnh khuôn mặt, tăng độ chính xác khi so sánh đặc trưng nhận diện.

- ***Hoạt hình khuôn mặt*** *(Face Animation):* Dùng để mô phỏng biểu cảm, chuyển động miệng, mắt, v.v. trong game, phim, hoặc avatar ảo.
- ***Phân tích đặc trưng khuôn mặt*** *(Facial Attribute Classification):* Xác định giới tính, độ tuổi, cảm xúc, tình trạng sức khỏe dựa trên các khu vực cụ thể trên khuôn mặt.
- ***Chỉnh sửa khuôn mặt*** *(Face Editing):* Ứng dụng trong làm đẹp, thay đổi cấu trúc khuôn mặt hoặc hiệu ứng AR trong thời gian thực.
- ***Hỗ trợ y tế và chăm sóc sức khỏe****:* Phát hiện dấu hiệu mệt mỏi, phân tích giấc ngủ, hoặc hỗ trợ người khiếm khuyết trong giao tiếp.

1.4 Phát biểu bài toán

Nhiệm vụ chính của định vị điểm mốc trên khuôn mặt là tìm chính xác vị trí của những điểm đặc trưng trên khuôn mặt, như là: khóe mắt, đường viền mắt, chóp mũi, khóe miệng, đường viền môi, chân mày, đường viền khuôn mặt [2].

Cụ thể, cho I là ảnh đầu vào, được biểu diễn dưới dạng tensor 3 chiều với kích thước $W \times H \times C$, trong đó W, H, C lần lượt là chiều rộng, chiều cao, và số kênh màu của ảnh. Thông thường, ảnh sẽ được biểu diễn dưới dạng một kênh màu (Binary Image, Gray Image) hoặc ba kênh màu (RGB Image).

Định vị điểm mốc trên khuôn mặt (Facial Landmark Localization) là bài toán tìm hàm [3]

$$\Theta : I \rightarrow Y$$

Từ ảnh đầu vào I dự đoán ma trận điểm mốc $\hat{Y} \in R^{N_L \times 2}$, trong đó:

- N_L là số lượng điểm mốc trên khuôn mặt.
- $\hat{Y}_{i1} \in [0; W]$ biểu diễn tọa độ X của điểm mốc thứ i.
- $\hat{Y}_{i2} \in [0; H]$ biểu diễn tọa độ Y của điểm mốc thứ i.

Số lượng điểm mốc trên khuôn mặt N_L và liên kết chính xác giữa điểm mốc trên khuôn mặt và vị trí của nó trên khuôn mặt (cái gọi là lược đồ chú thích) được xác định ở cấp độ tập dữ liệu. Ngoài ra, tập dữ liệu được sử dụng để xác định hình ảnh sử dụng huấn luyện hàm Θ (tập huấn luyện) và hình ảnh để đánh giá (tập kiểm tra).

Những độ đo được sử dụng để đánh giá chất lượng của mô hình như: Normalized Mean Error (NME, %), Failure Rate (FR, %), Cumulative Error Distribution Area Under Curve (CED-AUC).

1.5 Cấu trúc cơ bản của hệ thống định vị điểm mốc trên khuôn mặt

Phương pháp tiếp cận chung đối với bài toán định vị điểm mốc trên khuôn mặt gồm 2 bước chính là: **Face Detection** và **Landmark Localization** (như được minh họa trong Hình 1).

Hình 1: Facial landmark localization

- **Face Detection**: Xác định vị trí của khuôn mặt trong ảnh gốc (dưới dạng một hoặc nhiều hộp giới hạn - bounding boxes).
- **Landmark Localization**: Dự đoán tọa độ các điểm đặc trưng (landmarks) trên khuôn mặt đã được phát hiện.

1.6 Những thách thức của định vị điểm mốc trên khuôn mặt

- **Đa dạng hình dáng khuôn mặt** (*Facial Variability*): Mỗi người có cấu trúc khuôn mặt khác nhau về tỷ lệ mắt - mũi - miệng - cằm, v.v.. Bên cạnh đó, hình dáng khuôn mặt còn bị ảnh hưởng bởi độ tuổi, giới tính, chủng tộc.
- **Đa góc nhìn** (*Pose Variation*): Khuôn mặt có thể xoay các góc nghiêng khác nhau (yaw, pitch, roll).
- **Che khuất** (*Occlusion*): Khuôn mặt có thể bị che bởi các tác nhân khác nhau, nhưng mô hình vẫn cần phải dự đoán chính xác.
- **Điều kiện ánh sáng và chất lượng ảnh** (*Lighting and Image Quality*): Ảnh bị nhiễu, thiếu sáng, ngược sáng hoặc độ phân giải thấp có thể khiến mô hình không phát hiện được đúng khuôn mặt hay điểm mốc.

- **Yêu cầu thời gian thực** (*Real-time Inference*): Các ứng dụng thực tế của Facial Landmark Localization yêu cầu xử lý nhanh và chính xác trong thời gian thực. Nhưng việc cân bằng giữa tốc độ và độ chính xác vẫn là một thử thách lớn.

2 Chương 2: Cơ sở lý thuyết

2.1 Giới thiệu chung

Chương này trình bày các khái niệm và kiến thức nền tảng quan trọng trong lĩnh vực thị giác máy tính và học sâu, đặc biệt là những kỹ thuật cốt lõi được áp dụng trong các mô hình phát hiện điểm mốc khuôn mặt hiện đại. Việc hiểu rõ các cơ chế này sẽ tạo điều kiện thuận lợi cho việc tiếp cận nội dung của các chương sau, bao gồm tổng quan các công trình nghiên cứu liên quan (Chương 3) và phân tích sâu về phương pháp tiên tiến FaceXFormer (Chương 4).

2.2 Trích xuất đặc trưng trong thị giác máy tính

Trong thị giác máy tính, đặc trưng (feature) là những thông tin hữu ích được trích xuất từ dữ liệu hình ảnh đầu vào. Các đặc trưng này có thể là các cạnh, góc, vùng màu sắc, kết cấu, hoặc các biểu diễn phức tạp hơn được học bởi các mô hình học sâu. Việc trích xuất đặc trưng hiệu quả là bước tiền xử lý quan trọng, quyết định lớn đến hiệu suất của các tác vụ tiếp theo như nhận dạng, phát hiện đối tượng, hay phân đoạn ảnh.

2.2.1 Trích xuất đặc trưng đa tỷ lệ (Multi-scale Feature Extraction)

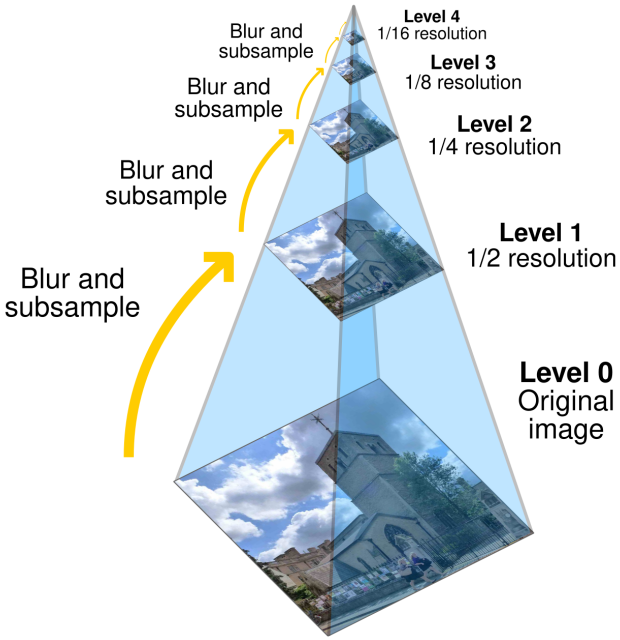
Các đối tượng trong một bức ảnh có thể xuất hiện ở nhiều kích thước và tỷ lệ khác nhau. Ví dụ, một khuôn mặt có thể chiếm phần lớn bức ảnh (tỷ lệ lớn) hoặc chỉ là một phần nhỏ (tỷ lệ nhỏ). Để mô hình có thể nhận diện và phân tích đối tượng một cách hiệu quả bất kể sự thay đổi về tỷ lệ, việc trích xuất đặc trưng đa tỷ lệ là vô cùng cần thiết.

Tầm quan trọng:

- **Thích ứng với kích thước đối tượng:** Giúp mô hình nắm bắt thông tin từ các đối tượng ở các khoảng cách và kích thước khác nhau.
- **Kết hợp ngữ cảnh cục bộ và toàn cục:** Đặc trưng ở tỷ lệ nhỏ thường chứa thông tin chi tiết (*local context*), trong khi đặc trưng ở tỷ lệ lớn hơn cung cấp thông tin ngữ cảnh rộng hơn (*global context*). Kết hợp cả hai loại thông tin này giúp mô hình hiểu sâu hơn về nội dung bức ảnh.
- **Tăng độ chính xác:** Nhiều nghiên cứu đã chỉ ra rằng việc sử dụng đặc trưng đa tỷ lệ giúp cải thiện đáng kể độ chính xác của các mô hình thị giác máy tính.

Phương pháp phổ biến:

Kim tự tháp ảnh (Image Pyramids): Là một kỹ thuật cổ điển, tạo ra nhiều phiên bản của ảnh gốc ở các độ phân giải khác nhau (xem Hình 2). Mô hình sau đó xử lý trên tất cả các phiên bản này.



Hình 2: Hình ảnh minh họa kim tự tháp ảnh với 5 mức độ phân giải khác nhau.

Mạng Kim tự tháp Đặc trưng (Feature Pyramid Networks - FPN): Trong các mạng nơ-ron tích chập (CNN), **FPN** [4] là một kiến trúc phổ biến để xây dựng các bản đồ đặc trưng đa tỷ lệ với chi phí tính toán hợp lý. FPN kết hợp các bản đồ đặc trưng từ các tầng sâu khác nhau của mạng (có độ phân giải khác nhau) để tạo ra một tập hợp các đặc trưng giàu thông tin ở nhiều tỷ lệ.

Kiến trúc Encoder đa tầng: Nhiều kiến trúc mạng hiện đại, đặc biệt là các kiến trúc dựa trên Transformer cho thị giác, thường có bộ mã hóa (encoder) được thiết kế để tự động trích xuất đặc trưng ở nhiều tầng khác nhau, mỗi tầng tương ứng với một mức độ chi tiết hoặc một tỷ lệ khác nhau. Ví dụ, trong phương pháp **FaceXFormer** [5], bộ mã hóa nhận ảnh đầu vào và trích xuất các đặc trưng đa tỷ lệ $F = \{F_1, F_2, F_3, F_4\}$, trong đó F_i là đặc trưng từ tầng thứ i. Các đặc trưng đa tỷ lệ này sau đó được hợp nhất bằng một module **MLP-Fusion** để thu được một biểu diễn khuôn mặt thống nhất. Cách tiếp cận này cho phép mô hình nắm bắt cả chi tiết nhỏ lẫn cấu trúc tổng thể của khuôn mặt.

2.3 Kiến trúc Transformer và Cơ chế Attention

Kiến trúc Transformer [6], ban đầu được giới thiệu bởi Vaswani và cộng sự vào năm 2017 cho các bài toán dịch máy trong xử lý ngôn ngữ tự nhiên (NLP), đã tạo ra một cuộc cách mạng và nhanh chóng được mở rộng ứng dụng sang nhiều lĩnh vực khác, bao gồm cả thị giác máy tính. Thành công của Transformer chủ yếu đến từ việc sử dụng cơ chế "attention", đặc biệt là "self-attention", cho phép mô hình cân nhắc tầm quan trọng của các phần khác nhau trong dữ liệu đầu vào.

2.3.1 Cơ chế Attention - Tổng quan

Cơ chế **attention** (chú ý) mô phỏng khả năng của con người trong việc tập trung vào những phần thông tin quan trọng nhất khi xử lý một lượng lớn dữ liệu. Thay vì xử lý tất cả thông tin đầu vào với vai trò như nhau, cơ chế attention cho phép mô hình gán các trọng số khác nhau cho các phần khác nhau của đầu vào, từ đó tập trung hơn vào những phần liên quan nhất đến tác vụ đang thực hiện.

Trong ngữ cảnh của Transformer, attention được mô tả thông qua ba thành phần chính:

- **Truy vấn (Query - Q):** Đại diện cho yếu tố hiện tại đang cần được xử lý hoặc tìm kiếm thông tin liên quan.
- **Khóa (Key - K):** Đại diện cho các yếu tố trong tập dữ liệu đầu vào mà Truy vấn sẽ so sánh để xác định mức độ liên quan.
- **Giá trị (Value - V):** Đại diện cho nội dung thông tin của các yếu tố đầu vào. Khi một Khóa được xác định là liên quan đến Truy vấn, Giá trị tương ứng của nó sẽ được sử dụng để tính toán đầu ra.

Nói một cách đơn giản, cơ chế attention tính toán "đầu ra" bằng cách lấy tổng có trọng số của các "Giá trị", trong đó trọng số của mỗi "Giá trị" được xác định bởi mức độ tương thích giữa "Truy vấn" tương ứng và "Khóa" của "Giá trị" đó.

2.3.2 Scaled Dot-Product Attention (Attention Tích Vô Hướng Có Điều Chỉnh Tỷ Lệ)

Đây là một dạng cụ thể của cơ chế attention được sử dụng rộng rãi trong các mô hình Transformer. Công thức chuẩn của Scaled Dot-Product Attention là:

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Trong đó:

- Q là ma trận các truy vấn (queries).
- K là ma trận các khóa (keys).
- V là ma trận các giá trị (values).
- d_k là chiều (số cột) của các vector khóa (và cũng là chiều của vector truy vấn).
- K^T là ma trận chuyển vị của K.

Giải thích chi tiết các bước tính toán:

1. **Tính điểm tương đồng (Similarity Scores):**

$$\text{Scores} = QK^T$$

Bước này tính toán tích vô hướng giữa mỗi vector truy vấn $q \in Q$ với tất cả các vector khóa $k \in K$. Nếu Q có kích thước $(N_q \times d_k)$ và K^T có kích thước $(d_k \times N_k)$, thì ma trận Scores sẽ có kích thước $(N_q \times N_k)$. Mỗi phần tử (i, j) trong ma trận Scores biểu thị mức độ tương đồng hoặc liên quan giữa truy vấn thứ i và khóa thứ j. Tích vô hướng càng lớn cho thấy sự tương đồng càng cao.

2. **Điều chỉnh tỷ lệ (Scaling):**

$$\text{Scaled Scores} = \frac{\text{Scores}}{\sqrt{d_k}}$$

Các điểm tương đồng sau đó được chia cho $\sqrt{d_k}$. Việc điều chỉnh tỷ lệ này rất quan trọng. Khi chiều d_k lớn, giá trị của tích vô hướng QK^T có thể trở nên rất lớn, đẩy các giá trị đầu vào của hàm softmax vào vùng có gradient rất nhỏ. Điều này làm cho quá trình huấn luyện trở nên khó khăn và chậm chạp. Việc chia cho $\sqrt{d_k}$ giúp ổn định gradient và quá trình huấn luyện.

3. **Áp dụng Hàm Softmax (Normalization):**

$$\text{Attention Weights} = \text{softmax}(\text{Scaled Scores})$$

Hàm softmax được áp dụng lên từng hàng của ma trận Scaled Scores. Nó chuyển đổi các điểm tương đồng đã được điều chỉnh tỷ lệ thành một phân phối xác suất, sao cho tổng các trọng số attention trên mỗi hàng bằng 1. Mỗi trọng số này (attention weight) thể hiện tầm quan trọng tương đối của mỗi giá trị (value) đối với một truy vấn cụ thể.

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

4. **Tính đầu ra có trọng số (Weighted Output):**

$$\text{Output} = \text{Attention Weights} \cdot V$$

Cuối cùng, ma trận Attention Weights được nhân với ma trận Giá trị V. Nếu Attention Weights có kích thước $(N_q \times N_k)$ và V có kích thước $(N_k \times d_v)$ (với d_v là chiều của các vector giá trị), thì ma trận Output sẽ có kích thước $(N_q \times d_v)$. Mỗi vector hàng trong ma trận Output là một tổng có trọng số của các vector giá trị, trong đó các trọng số chính là attention weights. Điều này có nghĩa là đầu ra cho mỗi truy vấn là một sự kết hợp của các giá trị, ưu tiên những giá trị có khóa tương thích cao với truy vấn đó.

2.3.3 Cơ chế Self-Attention (Tự Chú Ý)

Self-Attention, hay còn gọi là intra-attention, là một trường hợp đặc biệt của cơ chế attention nơi mà các Truy vấn (Q), Khóa (K), và Giá trị (V) đều bắt nguồn từ cùng một chuỗi đầu vào. Nói cách khác, mô hình cho phép mỗi vị trí trong chuỗi đầu vào "chú ý" đến tất cả các vị trí khác trong cùng chuỗi đó để tính toán biểu diễn cho chính nó.

Cách hoạt động: Cho một chuỗi đầu vào $X = (x_1, x_2, ..., x_n)$, Q, K, V thường được tạo ra bằng cách nhân X với ba ma trận trọng số có thể học được W^Q, W^K, W^V:

- $Q = XW^Q$
- $K = XW^K$
- $V = XWV$

Sau đó, các ma trận Q, K, V này được đưa vào công thức Scaled Dot-Product Attention như đã mô tả ở trên.

Mục đích:

- **Nắm bắt phụ thuộc nội tại:** Self-Attention giúp mô hình hiểu được mối quan hệ phụ thuộc giữa các phần tử khác nhau trong cùng một chuỗi đầu vào, bất kể khoảng cách giữa chúng. Ví dụ, trong một hình ảnh, nó có thể tìm ra mối liên hệ giữa các vùng khác nhau của đối tượng.
- **Biểu diễn theo ngữ cảnh:** Biểu diễn của mỗi phần tử trong chuỗi được cập nhật dựa trên ngữ cảnh của toàn bộ chuỗi, làm cho biểu diễn trở nên phong phú và chính xác hơn.

Self-Attention là thành phần cốt lõi trong các khối encoder và decoder của kiến trúc Transformer.

2.3.4 Multi-Head Self-Attention (Tự Chú Ý Đa Đầu)

Thay vì chỉ thực hiện một phép tính attention duy nhất, Multi-Head Attention cho phép mô hình "chú ý" đến thông tin từ các không gian con biểu diễn (representation subspaces) khác nhau tại các vị trí khác nhau một cách đồng thời. Điều này giống như việc nhìn vào một vấn đề từ nhiều góc độ khác nhau.

Cách hoạt động:

1. **Chiếu tuyến tính (Linear Projections):** Các ma trận Q, K, V ban đầu được chiếu tuyến tính h lần (với h là số lượng "đầu" attention) bằng các bộ ma trận trọng số W_i^Q, W_i^K, W_i^V khác nhau (và có thể học được) để tạo ra h bộ (Q_i, K_i, V_i).

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

   Mỗi phép chiếu này ánh xạ Q, K, V vào một không gian con có chiều nhỏ hơn.

2. **Attention song song:** Cơ chế Scaled Dot-Product Attention được áp dụng song song cho từng bộ (Q_i, K_i, V_i) trong số h đầu này, tạo ra h ma trận đầu ra $head_i$.

3. **Nối và Chiếu (Concatenation and Projection):** Các ma trận đầu ra $head_i$ từ h đầu sau đó được nối (concatenate) lại với nhau. Kết quả của việc nối này sau đó được chiếu tuyến tính một lần nữa thông qua một ma trận trọng số W^O (có thể học được) để tạo ra đầu ra cuối cùng của lớp Multi-Head Attention.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$$

Ưu điểm:

- **Khả năng học đa dạng:** Cho phép mỗi đầu attention tập trung vào các khía cạnh hoặc mối quan hệ khác nhau trong dữ liệu. Ví dụ, một đầu có thể tập trung vào mối quan hệ cú pháp ngắn hạn, trong khi một đầu khác tập trung vào mối quan hệ ngữ nghĩa dài hạn.
- **Tăng cường khả năng biểu diễn:** Bằng cách kết hợp thông tin từ nhiều đầu, mô hình có thể xây dựng các biểu diễn phong phú và mạnh mẽ hơn.

2.3.5 Cơ chế Cross-Attention (Chú Ý Chéo)

Cross-Attention là một biến thể của cơ chế attention nơi Truy vấn (Q) đến từ một chuỗi đầu vào, trong khi Khóa (K) và Giá trị (V) đến từ một chuỗi đầu vào khác. Điều này cho phép một chuỗi "chú ý" và tích hợp thông tin từ một chuỗi khác.

Cách hoạt động: Giả sử chúng ta có hai chuỗi đầu vào, X_1 và X_2.

- Truy vấn Q được tạo ra từ X_1 (ví dụ: $Q = X_1 W^Q$).
- Khóa K và Giá trị V được tạo ra từ X_2 (ví dụ: $K = X_2 W^K$, $V = X_2 W^V$).

Sau đó, công thức Scaled Dot-Product Attention được áp dụng như bình thường:

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Ứng dụng phổ biến:

- **Kiến trúc Encoder-Decoder:** Đây là ứng dụng điển hình nhất của Cross-Attention. Trong các mô hình dịch máy hoặc tạo chú thích ảnh, decoder (tạo ra chuỗi đầu ra) sẽ sử dụng Cross-Attention để "chú ý" đến các phần liên quan trong đầu ra của encoder (đã mã hóa thông tin từ chuỗi đầu vào). Truy vấn Q đến từ trạng thái ẩn của decoder, còn K và V đến từ các trạng thái ẩn đầu ra của encoder.
- **Hợp nhất thông tin từ các nguồn khác nhau:** Cross-Attention có thể được sử dụng để kết hợp thông tin từ các modality khác nhau (ví dụ: văn bản và hình ảnh) hoặc từ các loại biểu diễn khác nhau.

Trong **FaceXFormer** [5], cơ chế Cross-Attention được sử dụng một cách sáng tạo. Kiến trúc này giới thiệu một module **Bi-directional Cross-Attention (Chú ý Chéo Hai chiều)** trong

bộ giải mã FaceX. Module này xử lý đồng thời các token khuôn mặt (face tokens, chứa thông tin đặc trưng của khuôn mặt) và các token tác vụ (task tokens, đại diện cho các tác vụ phân tích khuôn mặt cụ thể như phát hiện điểm mốc). Cụ thể:

1. Các token khuôn mặt sẽ "chú ý" đến các token tác vụ để tạo ra các token khuôn mặt nhận biết tác vụ (task-aware face tokens).

2. Đồng thời, các token tác vụ sẽ "chú ý" đến các token khuôn mặt để tạo ra các token tác vụ nhận biết khuôn mặt (face-aware task tokens).

Cơ chế này cho phép mô hình học các biểu diễn mạnh mẽ và tổng quát hóa tốt hơn bằng cách cho phép tương tác và trao đổi thông tin hiệu quả giữa thông tin hình ảnh khuôn mặt và yêu cầu của từng tác vụ cụ thể.

2.4 Tổng quan học đa tác vụ

Học đa nhiệm (Multi-task learning - MTL) là một framework học máy trong đó nhiều tác vụ (tasks) được học đồng thời bằng cách tận dụng các biểu diễn dùng chung. Thay vì xem mỗi tác vụ một cách biệt lập, MTL cố gắng khai thác các điểm tương đồng và khác biệt giữa các tác vụ nhằm cải thiện khả năng tổng quát hoá. Về mặt hình thức, giả sử có T tác vụ, mỗi tác vụ có dữ liệu đầu vào X_t, nhãn Y_t, và hàm mất mát L_t. Học đa nhiệm tìm cách tối thiểu hoá một hàm mất mát tổng hợp:

$$\sum_{t=1}^{T} \alpha_t L_t(X_t, Y_t; \Theta)$$

Trong đó: Θ là các tham số mô hình học biểu diễn chia sẻ, α_t là các hệ số dùng để điều chỉnh tầm quan trọng tương đối của mỗi tác vụ.

2.4.1 Điểm mạnh

- Mô hình thường học được một biểu diễn ẩn dùng chung có lợi cho nhiều tác vụ. Việc chia sẻ tham số cho phép trích xuất các đặc trưng liên quan đến tất cả các tác vụ một cách hiệu quả.

- Khi có các tác vụ liên quan, việc kết hợp dữ liệu từ nhiều tác vụ giúp cải thiện hiệu suất trên những tác vụ có ít dữ liệu.

- Học đa nhiệm hữu ích khi cần xây dựng hệ thống cho nhiều miền tác vụ, ngôn ngữ cùng lúc, thay vì phải huấn luyện các mô hình riêng lẻ.

2.4.2 Thách thức

- Khi các tác vụ không liên quan chặt chẽ, việc chia sẻ tham số có thể làm giảm hiệu suất thay vì cải thiện.
- Việc cân bằng các đối số Θ trong hàm mất mát tổng thể là không đơn giản.
- Khi số tác vụ tăng, phải điều chỉnh hàm mất mát tổng thể, do đó, phải thực hiện huấn luyện lại từ đầu.

3 Chương 3: Các công trình nghiên cứu liên quan

3.1 Giới thiệu chung

Theo thời gian, các phương pháp phát hiện điểm mốc khuôn mặt (Facial Landmark Detection - FLD) đã trải qua nhiều giai đoạn phát triển, từ các thuật toán dựa trên mô hình thống kê đơn giản đến các phương pháp phức tạp sử dụng mạng nơ-ron sâu và gần đây là kiến trúc transformer. Sự tiến bộ này không chỉ cải thiện độ chính xác của việc phát hiện điểm mốc mà còn mở rộng khả năng ứng dụng của FLD trong các điều kiện thực tế đa dạng và phức tạp, chẳng hạn như hình ảnh có tư thế lớn, che khuất, hoặc ánh sáng không lý tưởng.

Trong chương này, chúng ta sẽ khám phá sự phát triển của các phương pháp FLD qua các thời kỳ, bắt đầu từ những phương pháp ban đầu, sau đó là các phương pháp dựa trên mạng nơ-ron với hai nhánh chính là hồi quy trực tiếp và hồi quy dựa trên heatmap, và cuối cùng là các phương pháp tiên tiến sử dụng kiến trúc transformer.

3.2 Các phương pháp ban đầu

Đặc điểm chung: Các phương pháp FLD ban đầu chủ yếu dựa trên các mô hình thống kê và kỹ thuật học máy truyền thống.

Các công trình nghiên cứu tiêu biểu:

Một trong những phương pháp tiên phong là **Active Shape Model (ASM)**, được giới thiệu bởi Cootes và Taylor vào năm 1995. ASM sử dụng một mô hình thống kê về hình dạng khuôn mặt, được biểu diễn bởi một tập hợp các điểm, và cố gắng điều chỉnh mô hình này để phù hợp với hình ảnh mới bằng cách tối ưu hóa vị trí của các điểm dựa trên đặc trưng hình ảnh cục bộ.

Tiếp theo, **Active Appearance Model (AAM)**, cũng do Cootes và Taylor phát triển, mở rộng ASM bằng cách kết hợp thông tin về kết cấu (texture) của khuôn mặt, cho phép mô hình hóa cả hình dạng và vẻ ngoài của khuôn mặt. AAM đã cải thiện độ chính xác so với ASM nhưng vẫn gặp khó khăn trong việc xử lý các biến đổi phức tạp như tư thế lớn hoặc biểu cảm đa dạng.

Constrained Local Model (CLM) là một phương pháp khác, sử dụng một tập hợp các bộ dò cục bộ cho mỗi điểm mốc, với các ràng buộc từ một mô hình hình dạng để đảm bảo tính nhất quán. CLM đã cho thấy hiệu quả tốt hơn trong một số trường hợp so với ASM và AAM.

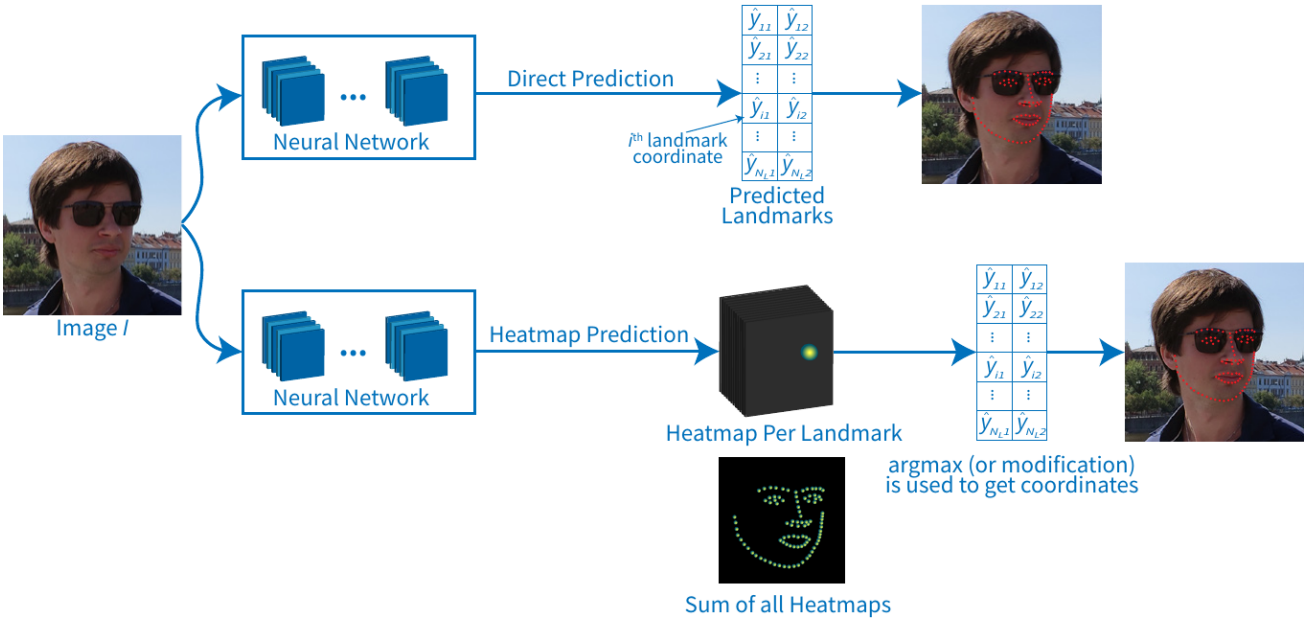
Một phương pháp đáng chú ý là **Ensemble of Regression Trees (ERT)** [7], được triển khai trong thư viện Dlib. ERT sử dụng gradient boosting để huấn luyện một tập hợp các cây

hồi quy, dự đoán vị trí của các điểm mốc từ đặc trưng HOG (Histogram of Oriented Gradients). ERT nhanh và hiệu quả, với thời gian xử lý khoảng 1 ms cho mỗi khuôn mặt, nhưng vẫn không đạt được độ chính xác cao trong các điều kiện khó khăn như tư thế lớn hoặc che khuất.

Hạn chế: Tuy nhiên, các phương pháp này đều có hạn chế khi đối mặt với các hình ảnh "in-the-wild", tức là các hình ảnh có sự che khuất, tư thế lớn, hoặc điều kiện ánh sáng không lý tưởng. Để khắc phục những hạn chế này, các phương pháp dựa trên học máy, đặc biệt là học sâu, đã được phát triển.

3.3 Các phương pháp dựa trên mạng nơ-ron

Với sự phát triển của học sâu, các phương pháp FLD hiện đại chủ yếu sử dụng mạng nơ-ron sâu, mang lại những cải tiến đáng kể về độ chính xác và khả năng xử lý các điều kiện phức tạp. Các phương pháp này có thể được chia thành hai nhánh chính (xem Hình 3): **hồi quy trực tiếp** (direct regression) và **hồi quy dựa trên heatmap** (heatmap-based regression).



Hình 3: Hàng trên mô tả phương pháp hồi quy điểm mốc trực tiếp: Bài toán được giải quyết dưới dạng hồi quy, trong đó tọa độ thực tế của các điểm mốc (x; y) được thuật toán dự đoán trực tiếp. Hàng dưới mô tả phương pháp hồi quy dựa trên heatmap: Thuật toán dự đoán phân bố xác suất vị trí của các điểm mốc dưới dạng heatmap. Một heatmap được tạo cho mỗi điểm mốc. Hàm argmax (hoặc biến thể của nó) được sử dụng để lấy tọa độ của từng điểm mốc.

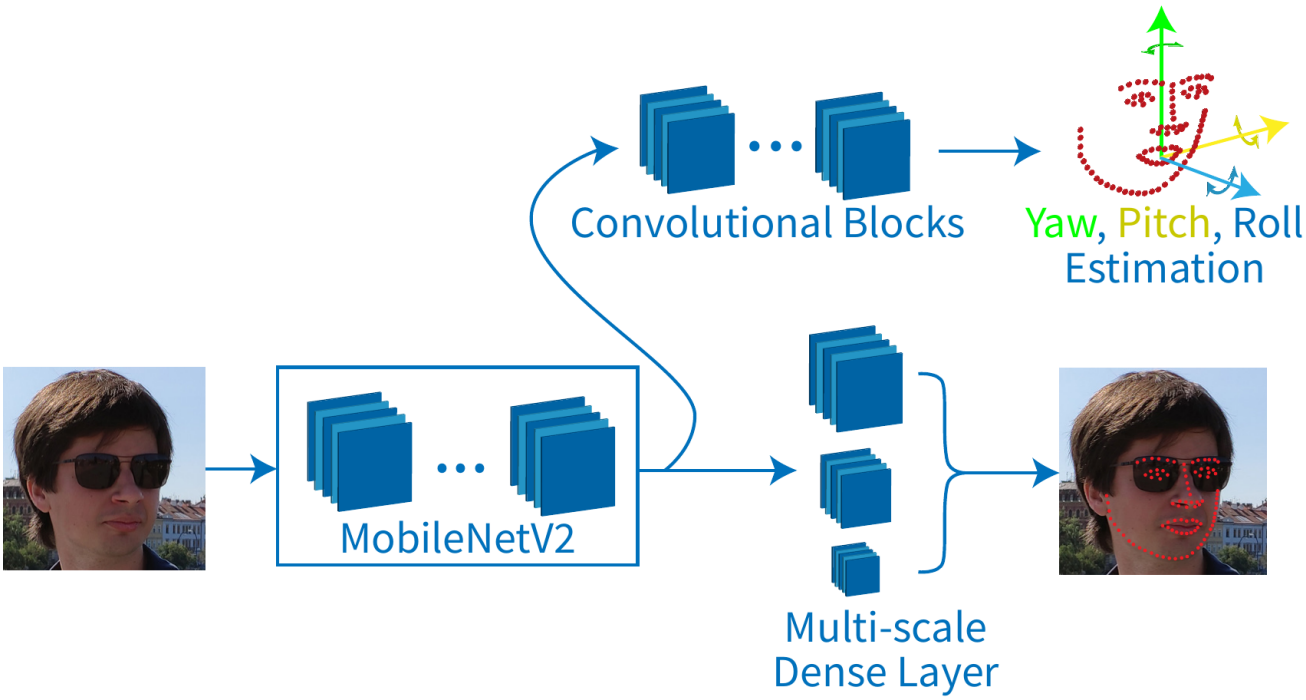
3.3.1 Phương pháp hồi quy trực tiếp

Đặc điểm chung: Các phương pháp hồi quy trực tiếp nhằm dự đoán trực tiếp tọa độ (x, y) của các điểm mốc từ đặc trưng hình ảnh. Những phương pháp này thường sử dụng mạng nơ-ron

tích chập (CNN) để trích xuất đặc trưng và sau đó áp dụng một hoặc nhiều lớp fully connected để dự đoán tọa độ.

Một số công trình nghiên cứu tiêu biểu:

Một ví dụ tiêu biểu là **Practical Facial Landmark Detector (PFLD)** [8], được thiết kế đặc biệt cho các thiết bị di động. PFLD sử dụng backbone MobileNetV2, một kiến trúc nhẹ, và bao gồm việc dự đoán góc khuôn mặt (yaw, pitch, roll) như một tác vụ phụ để cải thiện quá trình huấn luyện (xem Hình 4). PFLD đạt được sự cân bằng tốt giữa tốc độ và độ chính xác, phù hợp cho các ứng dụng thời gian thực trên thiết bị có tài nguyên hạn chế.



Hình 4: Kiến trúc của PFLD. MobileNetV2 được sử dụng làm bộ trích xuất đặc trưng với nhiều tác vụ: 1) để dự đoán vị trí điểm mốc khuôn mặt, lớp kết nối đầy đủ đa tỷ lệ được sử dụng, giúp thu nhận các đặc trưng hình ảnh ở nhiều tỷ lệ khác nhau tốt hơn (nhánh dưới); 2) các khối tích chập bổ sung được gắn vào MobileNetV2 để dự đoán góc quay khuôn mặt theo yaw, pitch, roll (nhánh trên). Các góc ước tính được nhúng vào hàm mất mát (loss) khi huấn luyện để cải thiện hiệu suất mạng tổng thể. Việc ước tính này không được thực hiện trong quá trình suy luận (inference) của mạng.

Một phương pháp khác là **Wing Loss** [9], giới thiệu một hàm mất mát mới để xử lý tốt hơn các giá trị ngoại lai, giúp cải thiện độ chính xác của dự đoán, đặc biệt là cho các điểm mốc có lỗi lớn. Phương pháp này đã được thử nghiệm với các backbone như CNN-6/7 và ResNet-50, và cải thiện thêm bằng kỹ thuật hard example mining (PDB).

Hạn chế: Tuy nhiên, các phương pháp hồi quy trực tiếp có thể gặp khó khăn trong việc đạt được độ chính xác cao, đặc biệt là trong việc định vị chính xác các điểm mốc ở mức pixel, do thiếu thông tin không gian chi tiết.

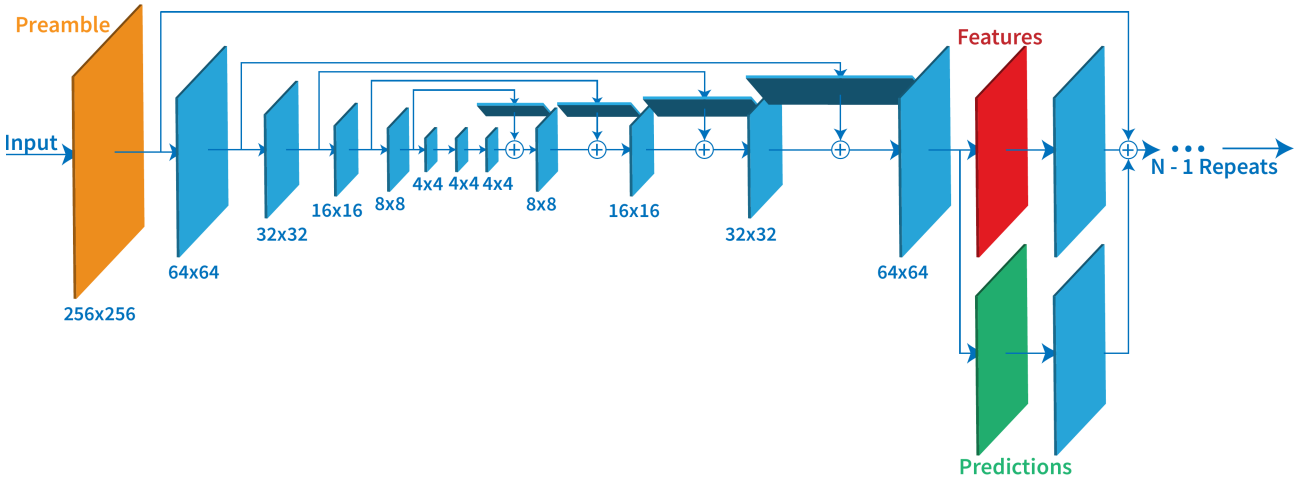
3.3.2 Phương pháp hồi quy dựa trên heatmap

Đặc điểm chung: Để khắc phục hạn chế của hồi quy trực tiếp, các phương pháp dựa trên heatmap đã được phát triển. Trong cách tiếp cận này, mạng nơ-ron dự đoán một heatmap 2D cho mỗi điểm mốc, trong đó giá trị cao nhất trên heatmap tương ứng với vị trí của điểm mốc. Các tọa độ điểm mốc sau đó được suy ra thông qua các kỹ thuật như argmax hoặc soft-argmax.

Các backbone thường dùng:

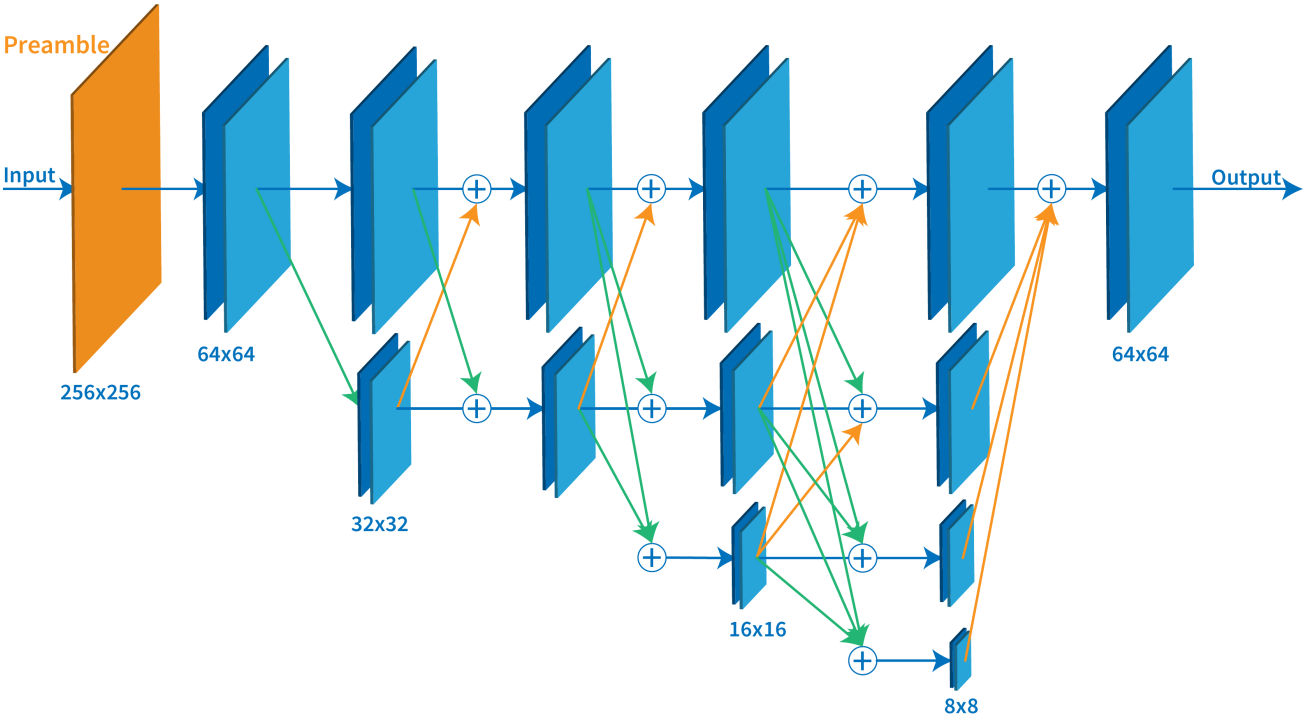
Các phương pháp này thường sử dụng các kiến trúc mạng như Hourglass, HRNet, và CU-Net. Những backbone này thường được thiết kế để duy trì thông tin không gian ở độ phân giải cao, điều rất quan trọng cho việc tạo ra các bản đồ nhiệt chính xác.

Mạng **Hourglass** [10] có kiến trúc đối xứng kiểu encoder-decoder (xem Hình 5). Ảnh đầu vào được xử lý qua các khối tích chập ở các độ phân giải đặc trưng khác nhau. Đầu tiên, độ phân giải bản đồ đặc trưng giảm dần sau mỗi khối tích chập (phần encoder), sau đó độ phân giải được khôi phục (tăng dần) sau mỗi khối (phần decoder). Việc xử lý ảnh ở nhiều độ phân giải giúp cải thiện độ chính xác cho nhiều tác vụ, bao gồm cả phát hiện điểm mốc. Kiến trúc **Stacked Hourglass** cho phép xếp chồng nhiều module Hourglass nối tiếp nhau (thường là 1, 2 hoặc 4 module). Mạng xuất ra các bản đồ nhiệt ở độ phân giải 64 $\times$ 64, mỗi điểm mốc có một bản đồ nhiệt riêng.



Hình 5: Mạng Hourglass có cấu trúc đối xứng với các khối downsampling và upsampling, cho phép trích xuất đặc trưng ở nhiều cấp độ và khôi phục chi tiết không gian.

Khác với các kiến trúc encoder-decoder truyền thống (như Hourglass) thường giảm độ phân giải rồi mới tăng lại, **HRNet (High-Resolution Network)** [11] duy trì các luồng đặc trưng có độ phân giải cao trong suốt quá trình xử lý (xem Hình 6). Nó bắt đầu với một luồng độ phân giải cao và dần dần thêm các luồng có độ phân giải thấp hơn song song. Quan trọng hơn, HRNet liên tục trao đổi thông tin giữa các luồng song song này, cho phép các đặc trưng ở độ phân giải cao được hưởng lợi từ ngữ cảnh rộng hơn của các đặc trưng ở độ phân giải thấp, và ngược lại. Việc duy trì độ phân giải cao và hợp nhất thông tin đa độ phân giải một cách hiệu quả giúp HRNet đạt được kết quả rất tốt trong các tác vụ yêu cầu độ chính xác không gian cao như phát hiện điểm mốc và ước lượng tư thế.



Hình 6: Kiến trúc tổng quát của HRNet.

Một số công trình nghiên cứu tiêu biểu:

Một phương pháp tiêu biểu là **Look at Boundary (LAB)** [12], kết hợp heatmap và hồi quy trực tiếp, sử dụng 4x Hourglass stack và giới thiệu boundary heatmaps như một biểu diễn trung gian để cải thiện việc định vị các điểm mốc. LAB cũng giới thiệu tập dữ liệu WFLW, một tập dữ liệu thử thách với các điều kiện đa dạng.

SubpixelHeatmap [13] là một phương pháp khác, sử dụng 2x Hourglass và local soft-argmax để giảm lỗi lượng tử hóa trong dự đoán heatmap, đạt được hiệu suất state-of-the-art trên nhiều tập dữ liệu như 300W, AFLW, và WFLW.

Các phương pháp khác đáng chú ý bao gồm:

- **Style Aggregated Network (SAN)** [14]: Sử dụng ResNet-152 và GANs (CycleGAN) để chuẩn hóa phong cách hình ảnh, cải thiện hiệu suất trong các điều kiện ánh sáng và màu sắc đa dạng.
- **Adaptive Wing Loss (AWing)** [15]: Xây dựng trên Wing Loss, sử dụng heatmap với hàm loss có thể vi phân để tạo ra các heatmap sắc nét hơn.
- **Geometry Aggregated Network (GEAN)** [16]: Sử dụng 4x Hourglass và các cuộc tấn công đối kháng trong quá trình huấn luyện để cải thiện độ chính xác.

Bảng dưới đây so sánh một số phương pháp dựa trên mạng nơ-ron:

Bảng 1: So sánh các phương pháp dựa trên mạng nơ-ron

Phương pháp	**Loại**	**Backbone**	**Đặc điểm nổi bật**
PFLD	Hồi quy trực tiếp	MobileNetV2	Nhẹ, dự đoán góc khuôn mặt
Wing Loss	Hồi quy trực tiếp	ResNet-50	Hàm loss cải tiến cho outliers
LAB	Hồi quy kết hợp	4x Hourglass	Boundary heatmaps, tập WFLW
SubpixelHeatmap	Hồi quy heatmap	2x Hourglass	Giảm lỗi lượng tử hóa
SAN	Hồi quy heatmap	ResNet-152	Chuẩn hóa phong cách hình ảnh
AWing	Hồi quy heatmap	Hourglass	Hàm loss vi phân

Ưu và nhược điểm: Các phương pháp dựa trên heatmap thường mang lại độ chính xác cao hơn so với hồi quy trực tiếp, nhưng có thể tốn kém hơn về mặt tính toán, đặc biệt khi sử dụng các kiến trúc phức tạp như Hourglass stack.

3.4 Các phương pháp tiên tiến và kiến trúc transformer

Kiến trúc Transformer, ban đầu được giới thiệu cho các tác vụ xử lý ngôn ngữ tự nhiên, đã nhanh chóng cho thấy tiềm năng to lớn trong lĩnh vực thị giác máy tính. Khả năng của Transformer trong việc mô hình hóa các mối quan hệ tầm xa (*long-range dependencies*) thông qua cơ chế self-attention đã mở ra những hướng tiếp cận mới cho nhiều bài toán, bao gồm cả phát hiện điểm mốc trên khuôn mặt. Một trong những xu hướng là xây dựng các **mô hình hợp nhất** có khả năng thực hiện nhiều tác vụ trong một kiến trúc duy nhất.

Một ví dụ nổi bật là **FaceXFormer** [5], một **mô hình transformer hợp nhất**, end-to-end, có khả năng thực hiện đồng thời **mười tác vụ phân tích khuôn mặt** khác nhau trong một framework duy nhất. FaceXFormer sử dụng kiến trúc encoder-decoder dựa trên transformer, trong đó mỗi tác vụ được biểu diễn bởi một token có thể học được. Điều này cho phép mô hình xử lý đồng thời nhiều tác vụ và học được các biểu diễn khuôn mặt tổng quát và mạnh mẽ.

Các thành phần chính trong kiến trúc của FaceXFormer:

Encoder: Trích xuất đặc trưng đa tỷ lệ từ hình ảnh khuôn mặt $I \in \mathbb{R}^{H\times W\times 3}$, tạo ra các đặc trưng phân cấp $\{S_i\}_{i=1}^{n}$ (với i từ 1 đến 4). Các đặc trưng này được hợp nhất bằng mô-đun MLP-Fusion để tạo biểu diễn khuôn mặt thống nhất F, với 983k tham số để đảm bảo hiệu suất thời gian thực.

Decoder (FaceX): Một decoder 2 tầng với cơ chế **cross-attention hai chiều**, xử lý cả token khuôn mặt F và token tác vụ $T = \langle T_1, \ldots, T_n \rangle$. Nó bao gồm: (1) **Task Self-Attention (TSA)** để tinh chỉnh token tác vụ bằng cách chú ý đến các token tác vụ khác; (2) **Task-to-Face Cross-Attention (TFCA)** và (3) **Face-to-Task Cross-Attention (FTCA)** để thực hiện các tương tác giữa token tác vụ và đặc trưng khuôn mặt.

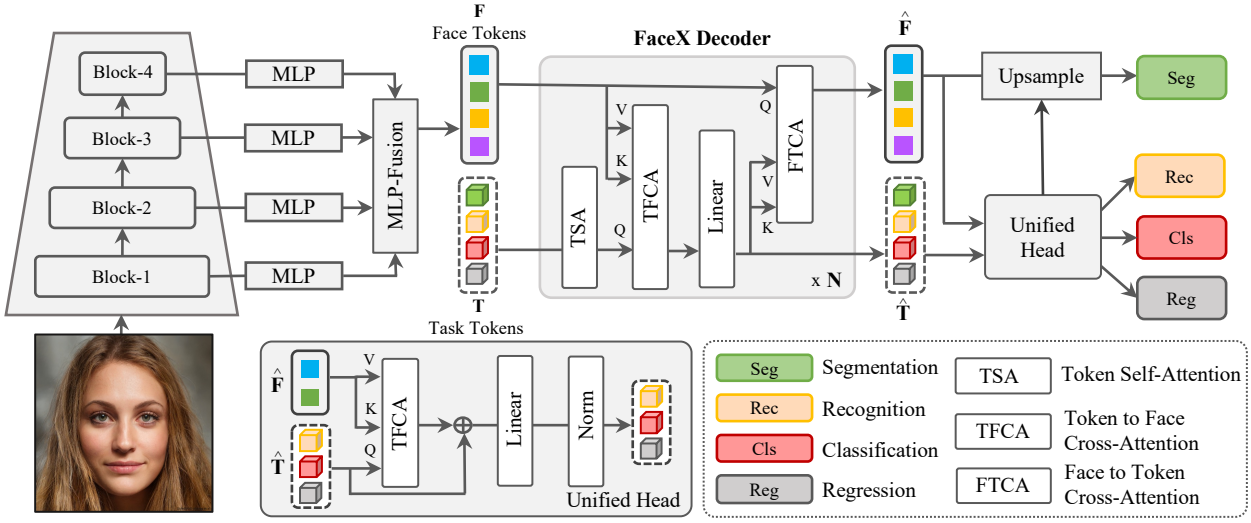
Unified Head: Xử lý các token tác vụ đầu ra thông qua các head cụ thể cho từng tác vụ. Đối với FLD, nó sử dụng mạng hourglass để dự đoán 68 điểm mốc.

Một số kết quả nổi bật: FaceXFormer đạt được hiệu suất state-of-the-art trên tập dữ liệu 300W với Normalized Mean Error (NME) là 3.05 (full), 2.66 (common), và 4.67 (challenge), vượt trội so với các mô hình trước như Faceptor (NME 3.16) và STARLoss (NME 3.97). Mô hình hoạt động ở tốc độ thời gian thực 33.21 FPS, với kích thước mô hình 109.29M tham số.

4 Chương 4: Phương pháp tiên tiến FaceXFormer

4.1 Tổng quan về framework của FaceXFormer

Framework của FaceXFormer được xây dựng dựa trên kiến trúc **encoder-decoder tiêu chuẩn** và được minh họa trong Hình 7:



Hình 7: Kiến trúc FaceXFormer

4.2 Multi-scale Encoder (Bộ mã hóa Đa tỷ lệ)

Mục đích: Các tác vụ phân tích khuôn mặt khác nhau đòi hỏi các loại đặc trưng khác nhau (ví dụ: ước tính tuổi cần biểu diễn toàn cục, trong khi phân vùng khuôn mặt cần biểu diễn chi tiết, v.v.). Và để giải quyết vấn đề này, nhóm tác giả đã đề xuất một bộ mã hóa đa tỷ lệ.

Quy trình: Mỗi ảnh đầu vào $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ được xử lý qua một tập hợp các lớp mã hóa. Mỗi lớp tạo ra các đặc trưng ở các mức độ trừu tượng và chi tiết khác nhau, tạo thành các đặc trưng đa tỷ lệ $\{\mathbf{S}_i\}_{i=1}^{n}$ (thường n=4). Các bản đồ đặc trưng $\mathbf{S}_i$ chuyển từ biểu diễn thô sơ đến chi tiết, phù hợp cho các tác vụ đa dạng.

4.2.1 Lightweight MLP-Fusion Module

Mục đích: Thay vì gán từng bản đồ đặc trưng $\mathbf{S}_i$ cho từng tác vụ (điều này sẽ không tối ưu), module này tạo ra một biểu diễn khuôn mặt hợp nhất $\mathbf{F}$ từ các đặc trưng đa tỷ lệ $\{\mathbf{S}_i\}_{i=1}^{n}$. Cách tiếp cận này hiệu quả hơn về số lượng tham số.

Quy trình:

1. Mỗi bản đồ đặc trưng $\mathbf{S}_i$ được đưa qua một lớp MLP riêng biệt để chuẩn hóa kích thước kênh (D_t).

$$\hat{\mathbf{S}}_i = \text{MLP}_{\text{proj}}(D_i, D_t)(\mathbf{S}_i), \forall i \in \{1, \ldots, n\}$$

2. Các đặc trưng đã biến đổi ($\hat{\mathbf{S}}_i$) được ghép nối.

$$\mathbf{F}_{\text{cat}} = \text{Concat}(\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, \ldots, \hat{\mathbf{S}}_n)$$

3. Kết quả ghép nối được đưa qua một lớp MLP hợp nhất (*fusion MLP*) để tạo ra biểu diễn hợp nhất $\mathbf{F}$.

$$\mathbf{F} = \text{MLP}_{\text{fusion}}(nD_t, D_t)(\mathbf{F}_{\text{cat}})$$

Trong đó: D_i là kích thước kênh của bản đồ đặc trưng thứ i; D_t là kích thước kênh đầu ra của lớp MLP (còn gọi là *kích thước mục tiêu*); và n là số lượng bản đồ đặc trưng. Kích thước đầu vào của lớp MLP hợp nhất là nD_t.

Ưu điểm: Theo nhóm tác giả, thiết kế gọn nhẹ của module này (chỉ có khoảng 983 nghìn tham số) giúp đảm bảo chi phí tính toán tối thiểu trong khi vẫn hợp nhất đặc trưng một cách hiệu quả, điều này rất quan trọng cho các ứng dụng thời gian thực.

4.3 Task Tokens (Token Tác vụ)

Ý tưởng: Lấy cảm hứng từ DETR [17], các tác giả giới thiệu khái niệm "**Task Tokens**" (*Token Tác vụ*). Mỗi tác vụ phân tích khuôn mặt (ví dụ: phân vùng, phát hiện điểm mốc, v.v.) được biểu diễn bằng một token riêng biệt, có thể học được, ký hiệu là: T. Hay nói cách khác, mỗi token tác vụ T là một biểu diễn nhúng tương ứng với một tác vụ phân tích khuôn mặt cụ thể.

Vai trò: Các token này cho phép mô hình xử lý đồng thời và hợp nhất nhiều tác vụ trong cùng một kiến trúc. Mỗi token được thiết kế để học các đặc trưng liên quan đến tác vụ cụ thể của nó từ biểu diễn khuôn mặt hợp nhất.

4.4 FaceX Decoder (FXDec)

Động lực: Các Transformer Decoder hiện có như DETR [17] hay Deformable-DETR [18] thường tốn kém về mặt tính toán, làm ảnh hưởng đến tốc độ xử lý của mô hình. Để khắc phục nhược điểm này, nhóm tác giả đã đề xuất một decoder mới, gọn nhẹ hơn, được gọi là **FaceX Decoder** (FXDec). Decoder này được thiết kế đặc biệt để mô hình hóa sự tương tác giữa các token tác vụ ($\mathbf{T} = \langle T_1, \ldots, T_n \rangle$) và token khuôn mặt ($\mathbf{F}$) một cách hiệu quả.

Cấu trúc: **FXDec** sử dụng cơ chế **cross-attention hai chiều** để tăng cường tương tác giữa token khuôn mặt và token tác vụ, bao gồm ba module con: Task Self-Attention (xem mục 4.4.1), Task-to-Face Cross-Attention (xem mục 4.4.2), và Face-to-Task Cross-Attention (xem mục 4.4.3) như được minh họa trong Hình 7.

Ưu điểm: Thiết kế gọn nhẹ của FaceX, với chỉ hai lớp decoder, giúp đạt hiệu suất thời gian thực bằng cách giảm độ phức tạp tính toán so với các decoder truyền thống như DETR [17] hoặc Deformable-DETR [18].

4.4.1 Task Self-Attention (TSA)

Mục đích: Module này được thiết kế để tinh chỉnh các biểu diễn đặc trưng cho từng tác vụ trong tập hợp các token tác vụ $\mathbf{T} = \langle T_1, \ldots, T_n \rangle$. Trong TSA, mỗi token T_i học cách chú ý đến các token tác vụ khác để nắm bắt các tương tác và mối quan hệ giữa các tác vụ, giúp học các đặc trưng không phụ thuộc vào tác vụ.

Hoạt động: Sử dụng cơ chế *multi-head self-attention* với Query, Key, Value đều là tập hợp các token tác vụ $\mathbf{T}$.

$$\mathbf{T}'_i = \text{SelfAttn}(\mathbf{Q} = T'_i, \mathbf{K} = \mathbf{T}, \mathbf{V} = \mathbf{T})$$

4.4.2 Task-to-Face Cross-Attention (TFCA)

Mục đích: Cho phép mỗi token tác vụ thu thập thông tin liên quan từ biểu diễn khuôn mặt hợp nhất $\mathbf{F}$.

Hoạt động: Sử dụng cơ chế *cross-attention*, với token tác vụ (*sau khi được tinh chỉnh ở module TSA*) đóng vai trò là Query, trong khi biểu diễn khuôn mặt hợp nhất ($\mathbf{F}$) đóng vai trò là Key và Value. Điều này cho phép mô hình trích xuất các đặc trưng quan trọng cho từng tác vụ.

$$\hat{T}_i = \text{CrossAttn}(\mathbf{Q} = T'_i, \mathbf{K} = \mathbf{F}, \mathbf{V} = \mathbf{F})$$

Đầu ra: Các task token sau khi được tinh chỉnh $\hat{\mathbf{T}} = \langle \hat{T}_1, \ldots, \hat{T}_n \rangle$.

4.4.3 Face-to-Task Cross-Attention (FTCA)

Mục đích: Tinh chỉnh biểu diễn khuôn mặt hợp nhất $\mathbf{F}$ dựa trên thông tin từ các token tác vụ đã được cập nhật ($\mathbf{T}'$).

Hoạt động: Sử dụng cơ chế *cross-attention*, với biểu diễn khuôn mặt $\mathbf{F}$ là Query, trong khi tập hợp các token tác vụ đã cập nhật ($\mathbf{T}' = \{\mathbf{T}''_1, \mathbf{T}''_2, \ldots, \mathbf{T}''_m\}$) là Key và Value. Cơ chế chú ý ngược này giúp tăng cường biểu diễn khuôn mặt bằng các chi tiết quan trọng liên quan đến tác vụ, hỗ trợ việc hợp nhất tác vụ.

$$\hat{\mathbf{F}} = \text{CrossAttn}(\mathbf{Q} = \mathbf{F}, \mathbf{K} = \mathbf{T}', \mathbf{V} = \mathbf{T}').$$

4.5 Unified-Head

Mục đích: Xử lý các token tác vụ ($\hat{\mathbf{T}}$) và token khuôn mặt ($\hat{\mathbf{F}}$) đã được tinh chỉnh từ FXDec để tạo ra các dự đoán cuối cùng cho từng tác vụ.

Quy trình: Dựa theo kiến trúc FaceXFormer ở Hình 7, các token đầu ra $\hat{\mathbf{F}}$ và $\hat{\mathbf{T}}$ tiếp tục được xử lý thông qua một module **TFCA** để tạo ra các đặc trưng được tinh chỉnh cuối cùng. Các token kết quả sau đó được đưa vào các "đầu" (*head*) tương ứng với từng tác vụ.

Các loại đầu (Task Heads): Kiến trúc của các đầu này khác nhau tùy theo bản chất của tác vụ:

- Phát hiện điểm mốc sử dụng mạng hourglass.
- Ước lượng tư thế đầu sử dụng MLP hồi quy.
- Nhận diện khuôn mặt sử dụng PartialFC [19].
- Các tác vụ như ước tính tuổi, giới tính và chủng tộc; nhận dạng biểu cảm khuôn mặt; dự đoán khả năng nhìn thấy của khuôn mặt; và dự đoán thuộc tính sử dụng các MLP phân loại.

- Phân vùng khuôn mặt liên quan đến việc *upsampling* $\hat{\mathbf{F}}$ và thực hiện *cross-product* với token phân vùng khuôn mặt để thu được bản đồ phân đoạn.

Số lượng Token: Số lượng token đầu ra cho mỗi tác vụ cũng khác nhau: Số lượng token cho phân đoạn tương ứng với tổng số lớp. Đối với dự đoán điểm mốc, số lượng token tương ứng với số điểm mốc (tức là 68). Đối với ước tính tư thế đầu, số lượng token là 9, đại diện cho ma trận xoay 3x3. Đối với các tác vụ khác, mỗi tác vụ sử dụng một token.

4.6 Multi-Task Training

FaceXFormer được huấn luyện đồng thời cho nhiều tác vụ phân tích khuôn mặt, tận dụng các bộ dữ liệu đa dạng. Quá trình này đối mặt với thách thức từ các yêu cầu tiền xử lý riêng biệt và đôi khi mâu thuẫn giữa các tác vụ (ví dụ: việc căn chỉnh điểm mốc có thể làm mất thông tin biến thiên của tư thế đầu).

Để giải quyết vấn đề này, FaceXFormer sử dụng các **token tác vụ đặc thù**. Chúng được thiết kế để trích xuất các đặc trưng liên quan đến từng tác vụ cụ thể từ một **biểu diễn khuôn mặt hợp nhất**. Cách tiếp cận này buộc mô hình phải học một biểu diễn tổng quát, mạnh mẽ, có khả năng hỗ trợ hiệu quả cho nhiều tác vụ khác nhau.

Quá trình huấn luyện nhằm tối ưu hóa một **hàm mất mát kết hợp**, là tổng có trọng số của các hàm mất mát riêng biệt, được điều chỉnh cho từng tác vụ:

$$L = \lambda_{seg}L_{seg} + \lambda_{lnd}L_{lnd} + \lambda_{hpe}L_{hpe} + \lambda_{attr}L_{attr} + \lambda_{a}L_{a} + \lambda_{g/r}L_{g/r} + \lambda_{exp}L_{exp} + \lambda_{fr}L_{fr} + \lambda_{vis}L_{vis}$$

Trong đó, mỗi thành phần L_{task} được lựa chọn phù hợp với bản chất của tác vụ (ví dụ: L_{seg} kết hợp Dice loss và Cross-Entropy loss cho phân vùng khuôn mặt; L_{lnd} sử dụng STAR loss cho dự đoán điểm mốc; L_{hpe} dùng Geodesic loss cho ước tính tư thế đầu; L_{fr} dùng ArcFace loss cho nhận dạng khuôn mặt; v.v.). Các trọng số λ_{task} được dùng để cân bằng tầm quan trọng và đóng góp của từng tác vụ vào quá trình huấn luyện tổng thể.

5 Chương 5: Dữ liệu và thực nghiệm

5.1 Thực nghiệm trong bài báo

5.1.1 Tập dữ liệu sử dụng

Quá trình huấn luyện và đánh giá mô hình xác định điểm mốc khuôn mặt thường được thực hiện trên một số bộ dữ liệu phổ biến. Dưới đây là những bộ dữ liệu nổi bật và thường được sử dụng trong các nghiên cứu in-the-wild.

1. **300W**
   - Gồm nhiều tập con (LFPW, AFW, HELEN, XM2VTS, IBUG) với tổng số 68 landmark/khuôn mặt.
   - Được chia thành các tập nhỏ để đánh giá: **Common**, **Challenge**, và **Full** (gồm toàn bộ ảnh).
   - Thích hợp để đánh giá khả năng nhận dạng mốc khuôn mặt trong điều kiện tương đối đa dạng (dáng pose trung bình, góc mặt vừa).
2. **AFLW**
   - Khoảng 25.000 ảnh với 21 landmark được gán nhãn.
   - Dữ liệu có độ đa dạng lớn về góc quay, có những ảnh quay nghiêng đến $\pm 120^\circ$ yaw, $\pm 90^\circ$ pitch.
   - Một số biến thể: AFLW-68 (được gán nhãn lại thành 68 điểm), hoặc Masked-AFLW (thêm mặt nạ).
3. **COFW (Caltech Occluded Faces in-the-Wild)**
   - Tập trung vào các khuôn mặt bị che khuất (occlusion) do tóc, tay, phụ kiện,...
   - Sở hữu 29 landmark cơ bản; có phiên bản COFW-68 (68 landmark) dùng để đánh giá.
4. **WFLW (Wider Facial Landmarks in-the-Wild)**
   - Số lượng landmark lớn (98 điểm), phong phú điều kiện chụp (pose lớn, makeup, biểu cảm mạnh, che khuất, ánh sáng kém, mờ nhoè, v.v.).
   - Chia thành các tập con gồm Pose, Expression, Illumination, Make-Up, Occlusion, Blur.
   - Hiện được xem là tập dữ liệu khó nhất, thường dùng để đánh giá các mô hình tiên tiến.

5.1.2 Các độ đo đánh giá

Trong bài toán xác định mốc khuôn mặt (facial landmark detection), các chỉ số đánh giá phổ biến gồm:

1. **Normalized Mean Error (NME, %)**

$$NME = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{N_L}\sum_{i=1}^{N_L}\frac{||Y_i - \hat{Y}_i||}{d}\right) \times 100$$

Trong đó:

- K là số ảnh trong tập test, N_L là số landmarks trên mỗi ảnh, $||Y_i - \hat{Y}_i||$ là khoảng cách Euclidean giữa vị trí mốc thực tế và vị trí dự đoán.
- d là hệ số chuẩn hóa, thưởng là khoảng cách hai khóe mắt (inter-ocular) hoặc khoảng cách hai đồng tử (inter-pupil).

NME càng **thấp** càng tốt.

2. **Failure Rate (FR, %)**

$$FR = \frac{1}{K}\sum_{k=1}^{K}[NME_k \geq 10\%] \times 100$$

FR càng **thấp** càng tốt.

3. **Cumulative Error Distribution (CED) & Area Under Curve (AUC)**

Đường cong biểu diễn tỉ lệ ảnh có lỗi nhỏ hơn một ngưỡng

Diện tích dưới đường cong (AUC) càng **lớn** càng tốt.

5.1.3 Thiết lập huấn luyện

Các mô hình được huấn luyện bằng cách sử dụng môi trường PyTorch phân tán trên tám GPU NVIDIA A6000 (48GB VRAM mỗi chiếc). Các backbone của mô hình được khởi tạo với trọng số đã huấn luyện trước trên ImageNet và xử lý ảnh đầu vào ở độ phân giải 224×224. Optimizer được sử dụng là AdamW, với hệ số weight decay là $1e^{-5}$.

Toàn bộ mô hình được huấn luyện trong 12 epoch, với batch size là 48 mẫu mỗi GPU. Learning rate khởi tạo là $1e^{-4}$, và giảm theo hệ số 10 tại các epoch thứ 6 và 10. Trong một số tác vụ, mô hình được huấn luyện thêm 3 epoch.

Augmentation:

Các phương pháp tăng cường dữ liệu được áp dụng ngẫu nhiên bao gồm: làm mờ Gaussian, chuyển sang ảnh grayscale, điều chỉnh gamma, che khuất, lật ngang, và các phép biến đổi affine (xoay, tịnh tiến, co giãn). Số lượng module FaceX decoder được đặt là 2.

Để đảm bảo quá trình huấn luyện ổn định khi sử dụng nhiều bộ dữ liệu với kích thước mẫu khác nhau, mỗi batch được cân bằng bằng cách lấy mẫu lại (upsampling) các bộ dữ liệu nhỏ hơn nhằm đảm bảo đại diện công bằng cho mỗi tác vụ.

5.1.4 Kết quả chính

Phần 4.3 trong bài báo [5] trình bày kết quả chính đạt được bởi mô hình FaceXFormer trên 10 tác vụ phân tích khuôn mặt. FaceXFormer đạt hiệu năng hàng đầu hoặc cạnh tranh trong hầu hết các tác vụ khi so sánh với cả các mô hình đơn nhiệm chuyên biệt và các mô hình đa nhiệm khác. Đáng chú ý:

- **Phân đoạn khuôn mặt (Face Parsing)**: Mô hình đạt mean F1 score 92.01% trên tập CelebAMaskHQ ở độ phân giải chỉ 224×224, vượt trội các mô hình khác dù dùng ảnh có độ phân giải gấp đôi.
- **Ước lượng tư thế đầu (Head Pose Estimation)**: Đạt $MAE = 3.52$, tốt hơn hầu hết các phương pháp chuyên biệt.
- **Phát hiện điểm đặc trưng (Landmark Detection)**: Đạt $NME = 4.67$, cải thiện so với Faceptor ($NME = 4.84$) và các mô hình khác.
- **Nhận diện biểu cảm (Expression Recognition)**: Đạt $accuracy = 88.24\%$, cao hơn nhiều mô hình đa nhiệm.
- **Dự đoán tuổi (Age Estimation)**: Đạt $MAE = 4.17$, chỉ thua một mô hình chuyên biệt ($MAE = 4.10$), đứng thứ hai.
- **Dự đoán thuộc tính và tầm nhìn khuôn mặt**: Accuracy lần lượt là 91.83% (CelebA) và 72.56% (COFW), đều cao nhất trong số các mô hình đa nhiệm.
- **Nhận dạng khuôn mặt (Face Recognition)**: Dù đạt kết quả cạnh tranh (mean accuracy = 95.94%), vẫn thấp hơn một số mô hình chuyên biệt (ví dụ: AdaFace đạt 97.41%). Điều này được lý giải do việc học đặc trưng không phụ thuộc danh tính khi huấn luyện đa nhiệm.

Điểm mạnh của FaceXFormer nằm ở khả năng hợp nhất 10 tác vụ phức tạp vào một mô hình duy nhất, hoạt động real-time với tốc độ 33.21 FPS, trong khi các mô hình khác chỉ đạt khoảng 14 FPS hoặc thấp hơn. Điều này thể hiện tính hiệu quả về tính toán và khả năng ứng dụng thực tế của mô hình.

5.1.5 Kết quả định tính trên dữ liệu thực tế ngoài tự nhiên

Phần 4.4 trong bài báo [5] minh họa chất lượng dự đoán của FaceXFormer qua hình ảnh thực tế ("in-the-wild"). Mô hình được thử nghiệm trên 4 ảnh đầu vào với nhiều điều kiện khó khăn như:

- Góc quay đầu lớn
- Làm mờ (blur)
- Che khuất (occlusion)

Kết quả cho thấy FaceXFormer có khả năng phân đoạn, định vị landmark, ước lượng tư thế đầu, phân loại tuổi, giới tính, chủng tộc và biểu cảm một cách chính xác, ngay cả trong điều kiện bất lợi. Điều này chứng minh tính robustness (khả năng chịu đựng biến thiên) và tổng quát hóa tốt của mô hình trong các ứng dụng thực tế.

Ngoài ra, khả năng xuất nhiều dạng nhãn (annotations) từ một ảnh duy nhất khiến FaceXFormer trở thành công cụ hữu ích cho các nhiệm vụ gán nhãn tự động (auto-annotation) trong các pipeline huấn luyện hoặc ứng dụng phân tích video, nhận dạng cá nhân, và giám sát.

5.2 Thực nghiệm của nhóm

5.2.1 Mô tả tập dữ liệu

Nhóm sử dụng tập dữ liệu iBUG 300W để đánh giá các phương pháp detect landmark khuôn mặt với 6000 ảnh huấn luyện (train), 666 ảnh validation và 1008 ảnh kiểm tra (test). Mỗi ảnh đều được chú thích với 68 điểm landmark. Seed sẽ được giữ cố định để đảm bảo tạo ra kết quả chạy nhất quán giữa các lần thực nghiệm.

5.2.2 Data Augmentation

Nhằm tăng cường sự đa dạng của tập dữ liệu huấn luyện, nhóm đã thực hiện quá trình tăng cường dữ liệu với các kỹ thuật như:

1. **Resize:** Chuẩn hóa kích thước ảnh về 224 x 224
2. **Random Affine:** Xoay ảnh ngẫu nhiên trong khoảng ±18°, Dịch ảnh ngẫu nhiên ±5% kích thước ảnh, căn chỉnh tỷ lệ khuôn mặt.
3. **Random Horizontal Flip:** Lật ảnh ngang ngẫu nhiên với xác suất 50%.
4. **Random Gray Scale:** Chuyển sang ảnh xám với xác suất 20%.
5. **Gaussian Blur:** Làm mờ ảnh với xác suất 30%.
6. **Gamma Adjustment:** Điều chỉnh độ sáng ngẫu nhiên với xác suất 20%.
7. **Random Erasing:** Mô phỏng che khuất với xóa ngẫu nhiên một vùng ảnh với các tỷ lệ khác nhau với xác suất 40%.

Với các tập kiểm định và kiểm tra (validation/test set) ta chỉ thực hiện chuẩn hóa dữ liệu với các phương thức như **Resize**, chuyển đổi về dạng Tensor và chuẩn hóa theo thống kê từ **ImageNet** để ổn định các giá trị pixels của hình.

5.2.3 Quá trình huấn luyện

Mục tiêu: Nhóm sẽ tiến hành thực nghiệm với 3 mục tiêu sau:

1. **Huấn luyện dựa trên Tọa độ Trực tiếp:** Mô hình dự đoán trực tiếp tọa độ (x, y) của các điểm mốc dưới dạng vector và sử dụng hàm mất mát **WingLoss** để tối ưu quá trình huấn luyện.
2. **Huấn luyện dựa trên Heatmap:** Mô hình dự đoán một heatmap (ma trận xác suất) cho mỗi điểm mốc và sử dụng hàm mất mát **STARLoss** để tối ưu quá trình huấn luyện.
3. **Thực nghiệm trên Pretrained Model:** Nhóm thực hiện đánh giá lại mô hình pretrain đã được tác giả cung cấp.

Thông số huấn luyện: Mô hình được huấn luyện trong **12** epochs với batch size **48**, learning rate ban đầu $\mathbf{1 \times 10^{-4}}$, weight decay $\mathbf{1 \times 10^{-5}}$, sử dụng bộ lập lịch `MultiStepLR` để giảm learning rate tại các epoch **6** và **10**.

Khởi tạo mô hình:

1. Sử dụng **Swin-B Transformer** (Swin-Base), đã được pretrained trên **ImageNet-22K**, làm backbone chính để trích xuất đặc trưng cho các tác vụ phát hiện điểm mốc.

2. Khởi tạo các tham số trong mô hình qua **Xavier Uniform** để cân bằng gradient giữa các tầng giúp hạn chế các trường hợp gradient vanishing/exploding.

Hàm mất mát:

1. **Coordinate Regression - WingLoss**

$$\mathcal{L}_{\text{Wing}} = \begin{cases} w \ln\left(1 + \frac{|y-\hat{y}|}{\epsilon}\right) & \text{, nếu } |y-\hat{y}| < w \\ |y-\hat{y}| - C & \text{, ngược lại} \end{cases}$$

   - $w = 10$: Ngưỡng chuyển đổi giữa chế độ log và linear
   - $\epsilon = 2$: Hệ số tỉ lệ trong hàm log, tránh chia cho 0
   - $C = w - w\ln(1 + w/\epsilon)$: Hằng số đảm bảo liên tục tại ngưỡng w

2. **Heatmap-Based - STARLoss**

$$\mathcal{L}_{\text{STAR}} = \mathcal{L}_{\text{dist}} + w \cdot \|\lambda\|_1$$

   - $\mathcal{L}_{\text{dist}}$: Hàm khoảng cách (Wing/SmoothL1) cho vị trí trung bình
   - λ càng lớn thì heatmap càng "phẳng", nghĩa là độ tin cậy thấp
   - $w = 1$: Trọng số cân bằng giữa loss vị trí và độ tin cậy

Tối ưu hóa:

1. **Optimizer:** Sử dụng **AdamW**

2. **Mixed Precision Training:** Sử dụng float16 cho phép tính forward/backward và float32 cho optimizer updates để giảm sử dụng bộ nhớ GPU và tăng tốc độ tính toán thông qua sử dụng PyTorch Native AMP với wrapper `torch.cuda.amp.autocast`

5.2.4 Kết quả thực nghiệm

Model	NME
Pretrained FaceXFormer	4.3664
FaceXFormer + SwinB + Coord + WingLoss	4.2830
FaceXFormer + SwinB + Heatmap + STARLoss	4.3846

Bảng 2: Kết quả thực nghiệm của nhóm

Kết quả thực nghiệm trên bộ dữ liệu IBug cho thấy mô hình **FaceXFormer + SwinB + Coord + WingLoss** đạt *hiệu suất tốt nhất* với chỉ số **4.2830**, vượt trội hơn so với mô hình **Pretrained FaceXFormer** (4.3664). Tương tự, mô hình **FaceXFormer + SwinB + Heatmap + STARLoss** cũng cho kết quả kém hơn (4.3846) cả 2 mô hình trên.

5.3 Ứng dụng mô hình vào thực tế

Ứng dụng mà nhóm đã xây dựng là sử dụng các điểm mốc trên khuôn mặt để gắn các hình dán lên khuôn mặt. Ứng dụng được xây dựng bằng Python, sử dụng thư viện Gradio để dựng giao diện web. Ứng dụng có 2 chức năng chính là:

1. Xác định, vẽ các điểm mốc lên khuôn mặt và gắn hình dán lên khuôn mặt sử dụng camera của thiết bị.

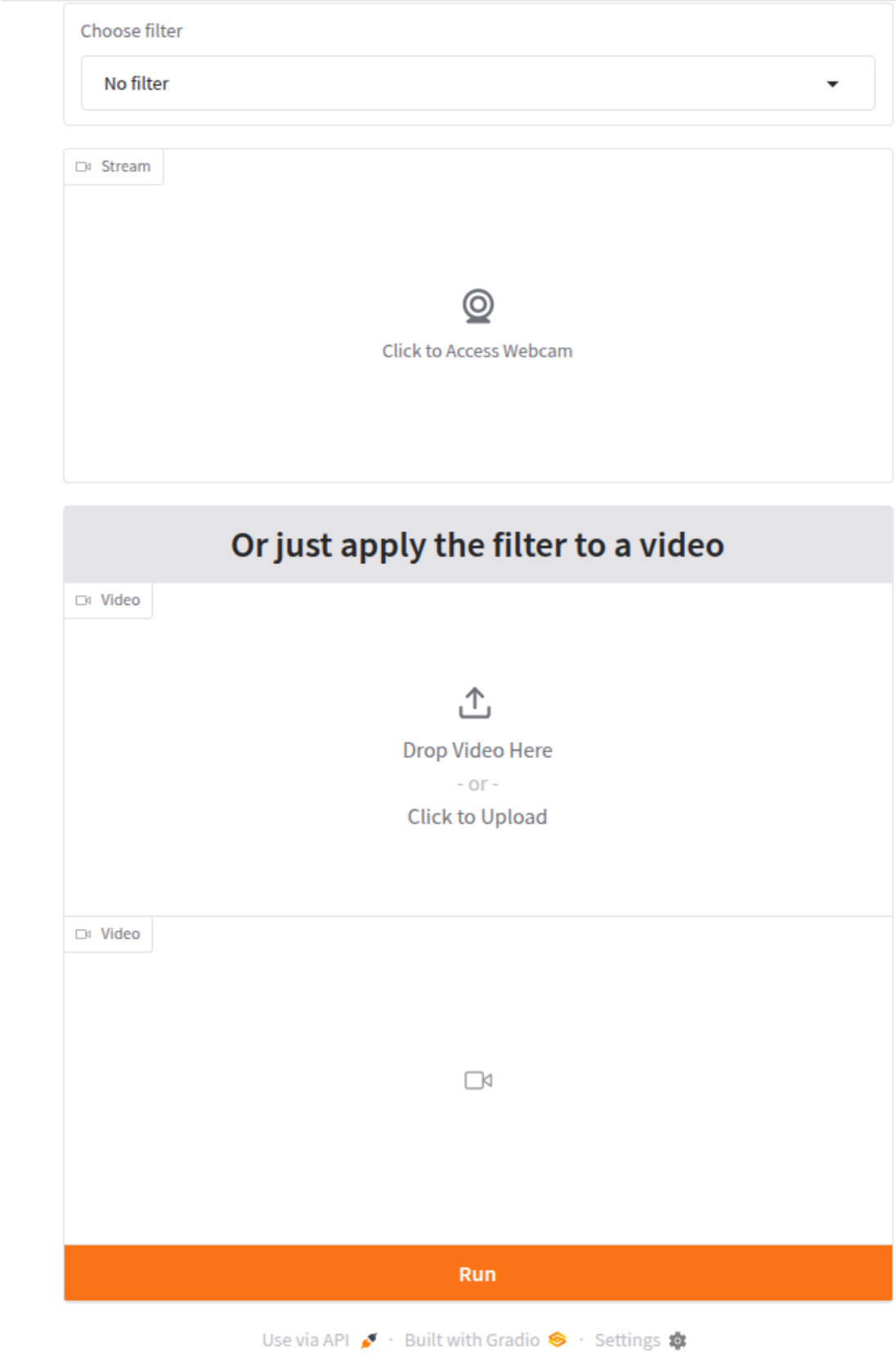
2. Xác định các điểm mốc lên khuôn mặt và gắn hình dán lên khuôn mặt trong video của người dùng.

Để gắn các hình dán lên mặt người dùng, ứng dụng sẽ chạy qua các bước sau:

1. **Xác định các khuôn mặt trong khung hình**: Ứng dụng sử dụng mô hình MTCNN được cung cấp bởi thư viện facenet-pytorch để xác định các khuôn mặt có trong khung hình.

2. **Xác định các điểm mốc trên từng khuôn mặt**: Ứng dụng đưa các khuôn mặt đã xác định qua mô hình FaceXFormer để xác định các điểm mốc trên khuôn mặt.

3. **Vẽ các điểm mốc lên khung hình**: Sau khi có được các điểm mốc, ứng dụng tiến hành vẽ các điểm mốc đó lên khung hình, mỗi điểm mốc được biểu diễn bằng một chấm tròn màu xanh dương.



Hình 8: Giao diện của ứng dụng

4. **Gắn hình dán lên khuôn mặt**: Ứng dụng gắn các hình dán lên các khuôn mặt trong khung hình sử dụng các điểm mốc đã được xác định. Các hình dán hiện tại sử dụng các điểm mốc ở khu vực mắt trái và phải để xác định vị trí và góc nghiêng.

5.4 Bàn luận

Trong thí nghiệm của nhóm, mô hình chỉ được huấn luyện trên một tác vụ là Facial Landmark Detection, khi đưa mô hình vào sử dụng trong ứng dụng thực tế, thì kết quả không như mong đợi. Mô hình hoạt động khá tệ đặc biệt khi khuôn mặt bị nghiêng. Một số nghiên cứu trước đó cũng chỉ ra vấn đề này [8, 20]. Điều này cho thấy tầm quan trọng của việc mô hình được học qua nhiều tác vụ.

6 Chương 6: Kết luận

Facial Landmark Localization (FLL) ngày càng khẳng định vị thế quan trọng trong nhiều lĩnh vực, từ nhận dạng khuôn mặt, phân tích biểu cảm, đến hỗ trợ y tế trong chẩn đoán và điều trị. So với các phương pháp truyền thống (dựa trên các mô hình thống kê và đặc trưng thủ công), *học sâu* đã mở ra bước nhảy vọt về độ chính xác lẫn tính linh hoạt, nhờ khả năng trích xuất và học đặc trưng từ lượng dữ liệu lớn một cách tự động. Tuy nhiên, thực tế vẫn tồn tại nhiều **thách thức**:

- **Độ đa dạng của môi trường và dữ liệu**: Sự thay đổi về góc quay, ánh sáng, biểu cảm, giới tính, độ tuổi hay sắc tộc đòi hỏi mô hình phải được huấn luyện trên dữ liệu phong phú và có tính đại diện cao.
- **Tính thời gian thực**: Ứng dụng trong các hệ thống giám sát an ninh hoặc thiết bị di động đòi hỏi tốc độ xử lý nhanh và ít tiêu tốn tài nguyên, trong khi vẫn duy trì độ chính xác.
- **Bài toán che khuất và biến dạng cực đoan**: Việc khuôn mặt bị che một phần, đeo khẩu trang, hay bị biến dạng do góc quay lớn khiến việc định vị điểm mốc gặp khó khăn, đòi hỏi giải pháp có khả năng suy luận dựa trên bối cảnh và đa nhiệm.
- **Quyền riêng tư và đạo đức**: Dữ liệu khuôn mặt mang tính cá nhân cao; do đó, yêu cầu khắt khe về mã hoá, ẩn danh hóa dữ liệu, cũng như tuân thủ các quy định pháp luật và chuẩn mực đạo đức.

Về lâu dài, **xu hướng phát triển** của lĩnh vực này là hướng tới các mô hình có khả năng *tự thích nghi* (adaptation) với môi trường và người dùng khác nhau, tích hợp cùng các nhiệm vụ liên quan (ví dụ: nhận diện cảm xúc, bệnh lý khuôn mặt, theo dõi cử động). Đồng thời, việc *giảm độ phức tạp mô hình* để phục vụ triển khai trên các thiết bị tính toán hạn chế cũng sẽ đóng vai trò quan trọng. Như vậy, **Facial Landmark Localization** vẫn là một mảng nghiên cứu năng động, đòi hỏi sự kết hợp chặt chẽ giữa xây dựng lý thuyết, thiết kế thuật toán, và cân nhắc về tính ứng dụng - hứa hẹn tiếp tục mang lại những bước tiến đáng kể cho cả cộng đồng học thuật lẫn công nghiệp.

Tài liệu tham khảo

[1] Xiaoqing Ding and Liting Wang. "Facial Landmark Localization". In: *Handbook of Face Recognition*. Ed. by Stan Z. Li and Anil K. Jain. London: Springer London, 2011, pp. 305–322. ISBN: 978-0-85729-932-1. DOI: 10.1007/978-0-85729-932-1_12. URL: https://doi.org/10.1007/978-0-85729-932-1_12.

[2] Stan Z. Li and Anil K. Jain. *Handbook of Face Recognition*. 2nd. Springer Publishing Company, Incorporated, 2011. ISBN: 085729931X.

[3] Kostiantyn Khabarlak and Larysa Koriashkina. "Fast Facial Landmark Detection and Applications: A Survey". In: *Journal of Computer Science and Technology* 22.1 (Apr. 2022), e02. ISSN: 1666-6046. DOI: 10.24215/16666038.22.e02. URL: http://dx.doi.org/10.24215/16666038.22.e02.

[4] Tsung-Yi Lin et al. *Feature Pyramid Networks for Object Detection*. 2017. arXiv: 1612.03144 [cs.CV]. URL: https://arxiv.org/abs/1612.03144.

[5] Kartik Narayan et al. *FaceXFormer: A Unified Transformer for Facial Analysis*. 2025. arXiv: 2403.12960 [cs.CV]. URL: https://arxiv.org/abs/2403.12960.

[6] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[7] Vahid Kazemi and Josephine Sullivan. "One millisecond face alignment with an ensemble of regression trees". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1867–1874. DOI: 10.1109/CVPR.2014.241.

[8] Xiaojie Guo et al. *PFLD: A Practical Facial Landmark Detector*. 2019. arXiv: 1902.10859 [cs.CV]. URL: https://arxiv.org/abs/1902.10859.

[9] Zhen-Hua Feng et al. "Wing Loss for Robust Facial Landmark Localisation with Convolutional Neural Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2235–2245. DOI: 10.1109/CVPR.2018.00238.

[10] Alejandro Newell, Kaiyu Yang, and Jia Deng. *Stacked Hourglass Networks for Human Pose Estimation*. 2016. arXiv: 1603.06937 [cs.CV]. URL: https://arxiv.org/abs/1603.06937.

[11] Ke Sun et al. "Deep High-Resolution Representation Learning for Human Pose Estimation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 5686–5696. DOI: 10.1109/CVPR.2019.00584.

[12] Wenyan Wu et al. "Look at Boundary: A Boundary-Aware Face Alignment Algorithm". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2129–2138. DOI: 10.1109/CVPR.2018.00227.

[13] Adrian Bulat, Enrique Sanchez, and Georgios Tzimiropoulos. *Subpixel Heatmap Regression for Facial Landmark Localization*. 2021. arXiv: 2111.02360 [cs.CV]. URL: https://arxiv.org/abs/2111.02360.

[14] Xuanyi Dong et al. "Style Aggregated Network for Facial Landmark Detection". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 379–388. DOI: 10.1109/CVPR.2018.00047.

[15] Xinyao Wang, Liefeng Bo, and Li Fuxin. "Adaptive Wing Loss for Robust Face Alignment via Heatmap Regression". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 6970–6980. DOI: 10.1109/ICCV.2019.00707.

[16] Seyed Mehdi Iranmanesh et al. "Robust Facial Landmark Detection via Aggregation on Geometrically Manipulated Faces". In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020, pp. 319–329. DOI: 10.1109/WACV45572.2020.9093508.

[17] Nicolas Carion et al. "End-to-End Object Detection with Transformers". In: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*. Glasgow, United Kingdom: Springer-Verlag, 2020, pp. 213–229. ISBN: 978-3-030-58451-1. DOI: 10.1007/978-3-030-58452-8_13. URL: https://doi.org/10.1007/978-3-030-58452-8_13.

[18] Xizhou Zhu et al. "Deformable DETR: Deformable Transformers for End-to-End Object Detection". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=gZ9hCDWe6ke.

[19] Xiang An et al. "Partial FC: Training 10 Million Identities on a Single Machine". In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2021, pp. 1445–1449. DOI: 10.1109/ICCVW54120.2021.00166.

[20] Zhanpeng Zhang et al. "Learning Deep Representation for Face Alignment with Auxiliary Attributes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.5 (May 2016), pp. 918–930. ISSN: 2160-9292. DOI: 10.1109/tpami.2015.2469286. URL: http://dx.doi.org/10.1109/TPAMI.2015.2469286.