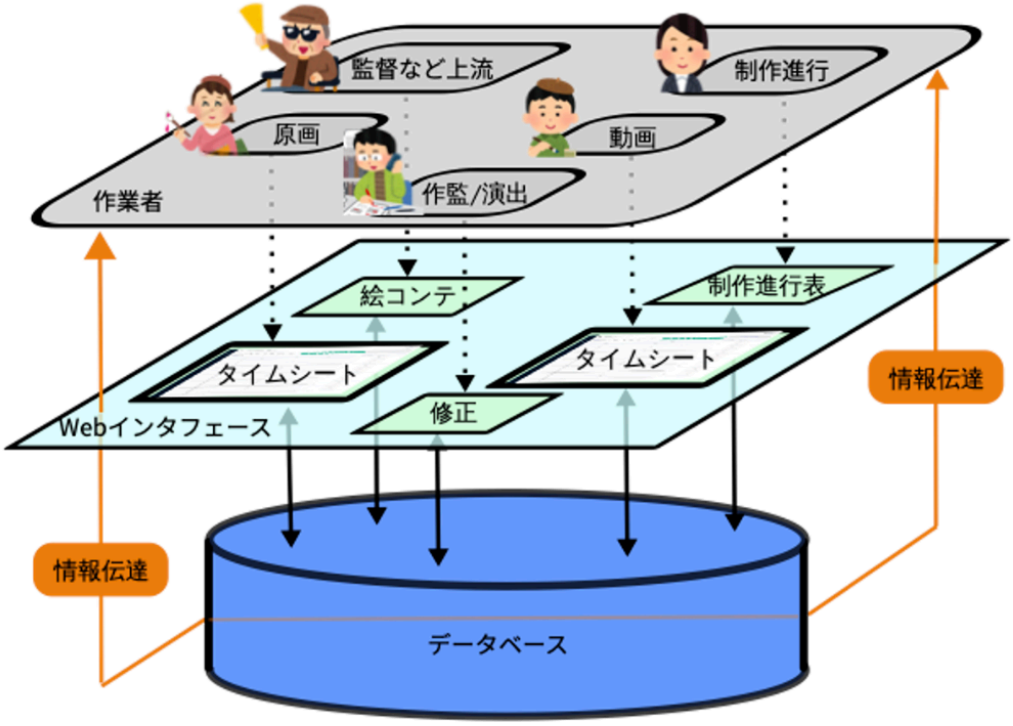

図 2.13　渕上らのシステム：全体設計

2.5　既存アニメ関連ツールの比較

2.5.1　比較指標

2.5.2　比較結果

2.6　タグ付けのための物体検出

本研究では、アニメーション制作過程の修正を活用しやすい形で蓄積するため、修正が入ったカット、特に彩色後のフレーム画像から自動的にタグを付与することを目的とする。その基盤技術として、画像中の対象（人物、顔、手、衣装、小物など）を領域（バウンディングボックス）とともに抽出できる物体検出は有用である。一方で、アニメ画像に対して十分な教師データ（バウンディングボックス付きデータセット）を用意することは困難であるため、本研究が物体検出器に求める要件は次の通りとする。

1. 学習データを用意せずに適用できる ゼロショット検出 が可能であること
2. 速度は重視しない（オフライン処理を想定）こと
3. アニメ画像に対して相対的に高い精度を示す手法を採用すること

以上を踏まえ、クローズドセット検出器（YOLO, DETR, DINO）と、言語条件により語彙を拡

張できるオープンボキャブラリ検出器（GLIP, Grounding DINO）を取り上げ、特に要件 (1)(3) を満たす候補として後者を中心に整理する。

2.6.1 YOLO [8]

YOLO は物体検出を代表する手法の一つであり、特に「画像を一度だけネットワークに通して検出を完結させる」単段（one-stage）検出器の流れを決定づけたモデルとして位置付けられる。YOLO（You Only Look Once）は Redmon らにより提案された CNN ベースの検出器で、入力画像から特徴を抽出するバックボーン CNN と、検出ヘッドを一体化して持つ点に特徴がある。図 2.14 に示すように、畳み込み層とプーリング層を中心に段階的に特徴マップを得たのち、最終段でバウンディングボックスの位置・サイズとクラス確率を同時に推定する。二段方式（候補領域生成して分類する方式）と比べて処理が単純であるため、推論が高速になりやすく、リアルタイム物体検出の文脈で広く普及した。

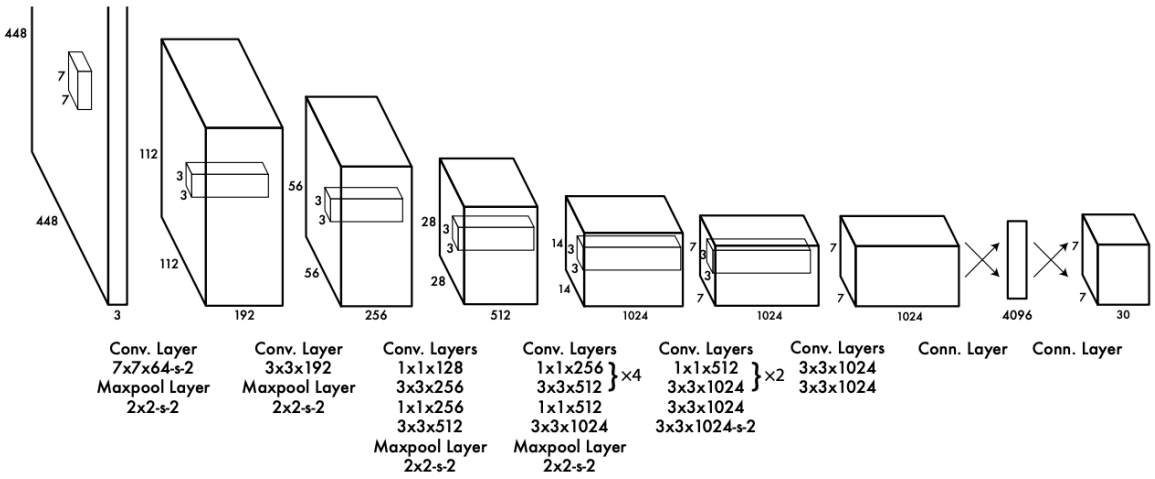

図 2.14　YOLOv1 のネットワーク構造の例

初期の YOLOv1 は速度面で大きな成功を収めた一方、小物体の見逃しや精度面に課題が指摘されていた。その後の改良として、YOLOv2 / YOLO9000 ではバッチ正規化、アンカーボックス、高解像度入力といった工夫により精度と再現率が大きく改善され、YOLOv3 ではより深いバックボーン設計とマルチスケール検出によって速度と精度のバランスがさらに洗練された。以降も YOLOv4 以降の派生や、Ultralytics 社による v5〜v12 など多くのバリエーションが提案されており、物体検出の実用モデルとして多様な応用で利用されている。

本研究の修正が入ったカットのタグ付けという文脈に照らすと、YOLO 系の利点は「多数フレームに対して候補領域を一括抽出する処理を比較的軽量に実行できる」点にある。制作現場の素材はフレーム数が膨大になり得るため、処理時間を抑えやすいこと自体は魅力である。ただし、本研究ではオフラインでの一括処理も想定でき、リアルタイム性は必須要件ではないため、高速性は採用判断を左右する決定要因というより、実装・運用上の余裕をもたらす副次的な利点として位置付けるのが自然である。

一方で、YOLO 系を自動タグ付けの中核として用いる際には、構造上の制約が生じる。多くの

YOLO 系モデルは COCO など自然画像データセットに基づくクローズドセット検出器として設計され、検出クラスはあらかじめ定義されたカテゴリ集合に限定される。このため、アニメ制作で検索や分析に有用な「上半身」「顔」「目」といった部位概念や、作品・工程ごとに柔軟に増やしたいタグ語彙を、学習データなしでそのままゼロショットに扱うことは原理的に難しい。加えて、アニメ画像（線画や彩色済み画像）は自然画像と統計的性質が異なるにもかかわらず、アニメ領域ではバウンディングボックス付きの大規模データセットが十分に整備されていない。その結果、YOLO をアニメ画像に対して安定して高精度に動作させるには追加学習やファインチューニングが必要になりやすいが、そのための教師データ準備コストが大きく、本研究が前提とする「学習データを用意しにくい状況でのタグ付け」という要件とは整合しにくい。

以上より、YOLO 系は「定義済みの少数クラスに対して高速に候補領域を抽出し、フレーム内の大まかな配置を把握する」といった用途では有効であり、物体検出手法の代表例として整理する価値がある。一方で、本研究が重視するゼロショット性とタグ語彙の柔軟性、そしてアニメ画像への相対的な精度という観点では、クローズドセット検出器である YOLO を主たる手法として採用することは難しい。したがって本章では、YOLO を物体検出の基礎的枠組みを示す参照（ベースライン）として位置付けつつ、後続で扱うオープンボキャブラリ検出器へと議論を接続する。

2.6.2　DETR [1]

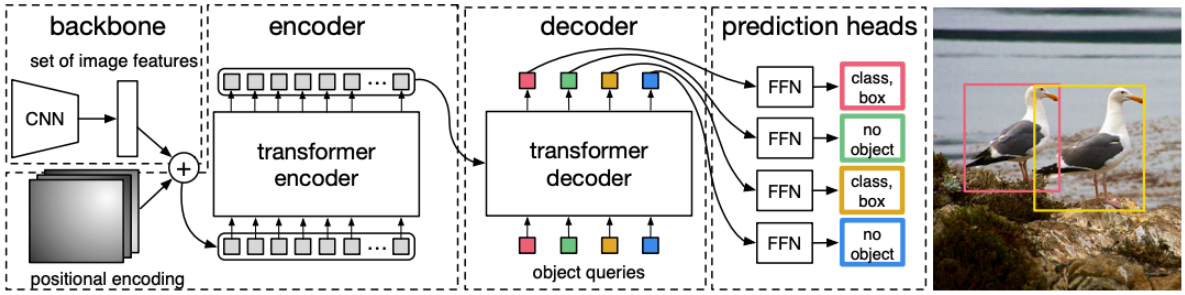

図 2.15　DETR の全体構成

DETR（DEtection TRansformer）は Carion らによって 2020 年に提案された物体検出モデルであり、従来主流であった「候補領域の生成→分類・回帰」という段階的パイプラインを、Transformer を中核とする単一のネットワークに統合した点に特徴がある。物体検出の研究史の中では、CNN を中心に発展してきた検出器設計に対して、Transformer を用いた 集合予測（set prediction）の枠組みを導入し、後続の多くの Transformer 系検出器（改良 DETR 群や DINO 系、さらには言語条件付き検出へ）につながる重要な転換点となったモデルとして位置付けられる。図 2.15 に DETR の全体構成を示す。

DETR の処理は大きくバックボーン、Transformer エンコーダ、Transformer デコーダの三段に整理できる。まず入力画像は CNN バックボーンにより特徴マップへ変換され、空間方向に並べ替えられたうえで位置エンコーディングが付与される。これが Transformer エンコーダへ入力され、自己アテンションによって画像全体の文脈を踏まえた特徴量が得られる。次に Transformer デコーダには固定個数のオブジェクトクエリ が入力される。オブジェクトクエリは学習可能なベクトル集合であり、画像中の各物体を「どのスロットが担当するか」を決めるための枠組みとして機能する。デ

コーダはクエリとエンコーダ出力の間でクロスアテンションを行うことで、画像中の重要領域をクエリごとに参照し、最終的に各クエリに対応する出力を prediction head に入力してクラスラベルとバウンディングボックスを推定する。従来の Anchor Base や非最大抑制 (NMS) に依存せず、ネットワーク全体を end-to-end に学習できるため、検出器の構造を比較的簡潔に記述できる点が利点として挙げられる。また、エンコーダの自己アテンションにより画像全体の関係性を扱えることから、物体同士の重なりや離れた領域間の関係が重要になる場面で有利に働く可能性がある。

一方で、元の DETR には実用面での課題も報告されている。代表的には、学習の収束が遅く、COCO のような一般的データセットでも十分な精度に到達するまでに多くの学習エポックを要する点が挙げられる。また、高解像度画像における小物体の検出性能が十分でないことが指摘されており、アニメ制作の文脈で頻出する小さな顔パーツや小物、アクセサリ類の検出では不利になり得る。さらに Transformer を中核とする性質上、軽量な CNN ベースの検出器と比べて推論時間やメモリ使用量が大きくなりやすく、計算資源の観点で工夫が必要となる場合がある。ただし本研究では速度を重視しない運用が想定されるため、計算コストそのものは致命的な制約というより、候補モデル選択におけるトレードオフ要因として整理するのが適切である。

本研究との関係で整理すると、DETR は物体検出を Transformer による end-to-end な枠組みとして定式化した点で重要であり、後続の改良手法（DINO など）や、言語条件を導入した検出器（Grounding DINO など）を理解するための基礎として有用である。一方で、DETR 自体は基本的にクローズドセット検出であり、アニメ画像に対する教師データが十分に用意できない状況で学習なしにタグ語彙を柔軟に増やしながら検出するという本研究の要件であるゼロショット性を満たしていない。したがって本章では、DETR を Transformer 系検出器の代表例・歴史的起点として位置付けつつ、収束性や小物体検出といった課題がどのように改善され、さらにゼロショット検出へ発展していくかという流れの中で次節以降の手法へ接続する。

2.6.3 DINO [9]

DINO は Zhang らによって提案された Transformer ベースの物体検出モデルであり、DETR 系列の課題であった学習収束の遅さと小物体検出性能の不足を改善することを目的としている。図 2.16 に DINO の全体構成を示す。

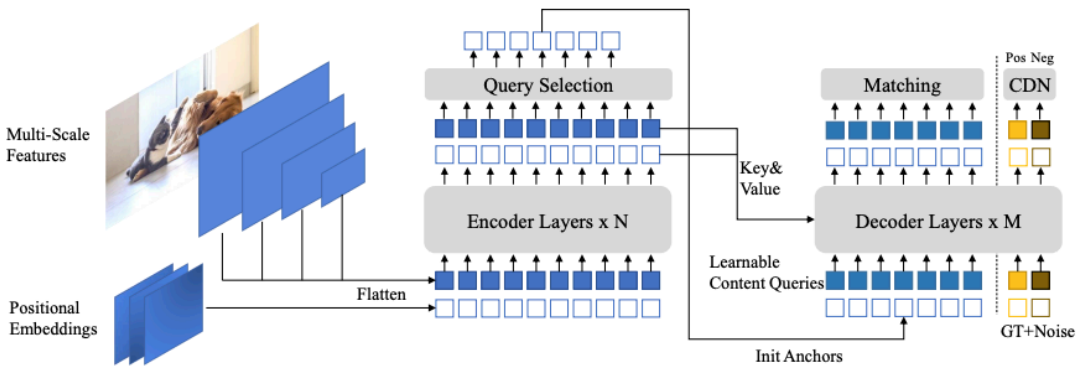

図 2.16　DINO の全体構成図

まず、入力画像からは CNN バックボーンによりマルチスケールの特徴マップが抽出され、それぞれに位置エンコーディングを加えた上でフラット化し、Transformer エンコーダに入力する。エンコーダは複数スケールの特徴を統合しつつ自己アテンションにより文脈情報を取り込んだ画像特徴を生成する。

DETR ではオブジェクトクエリがランダム初期化された学習パラメータとして与えられていたのに対して、DINO ではエンコーダ出力から「クエリ選択（Query Selection）」を行い、物体らしさの高い位置を初期アンカーとして抽出する。これらのアンカー情報をもとに内容ベクトル（Content Query）を生成し、Transformer デコーダに入力することで、各クエリが画像中の有望な領域に対応しやすいように設計されている。デコーダの出力は DETR 同様にクラスラベルとバウンディングボックスを予測するネットワークに入力され、最終的な検出結果として解釈される。

また DINO では、学習の安定化と性能向上のために「Contrastive DeNoising（CDN）」と呼ばれる学習戦略を導入している。これは、正解ボックスにノイズを加えた擬似ターゲットと、全く関係のないネガティブなボックスを同時にデコーダへ入力し、どの出力が真の物体に対応するかを学習させる手法である。このノイズ付きターゲット学習により、モデルはローカライゼーション誤差や外れ値に対して頑健になり、少ないエポックでも高い性能に到達しやすくなると報告されている。

このように DINO は、DETR のエンコーダ・デコーダ構造を継承しつつ、クエリ選択による初期アンカー生成とノイズ付き学習戦略を組み合わせることで、収束の高速化と小物体を含む高精度な検出を実現したモデルである。一方で、Transformer を中核とする点は DETR と同様であり、YOLO 系の軽量な CNN ベース検出器と比較すると推論時の計算コストは依然として大きい。またゼロショット性を備えていないため、アニメ制作で欲しい「顔」「手」「上半身」といった部位概念や、作品・工程依存で増やしたいタグ語彙を学習なしで柔軟に扱う主手法としては限界がある。

2.6.4 GLIP [4]

GLIP は Li らによって 2022 年に提案された物体検出モデルであり、画像と言語を統合的に扱う点に特徴がある。従来の物体検出では、画像と、それに対応するバウンディングボックスとクラスラベルの組からなる検出データのみを用いて学習するのが一般的であったのに対して、GLIP ではこれに加えて、文章中のフレーズと画像内の領域を対応付けたグラウンディング用データや、画像とキャプションのペアからなる画像テキストデータを併用して事前学習を行う。具体的には、物体検出データセット上では従来と同様に「ボックス＋クラス名」の組を学習しつつ、グラウンディングデータセット上では「文章中のフレーズ」と「それに対応する画像領域」の対応関係を学習し、さらに画像全体とキャプション文のペアからは、画像特徴とテキスト特徴が意味的に対応するように事前学習を行っている。このように複数形式の画像テキストデータを統合的に用いることで、GLIP は言語に敏感な物体表現を獲得し、ゼロショット設定においても高い検出性能を示すことが報告されている。

図 2.17 に GLIP の全体構成を示す。上段では「person」「bicycle」「hairdryer」などのカテゴリ名や簡単なフレーズからなるプロンプトをテキストエンコーダで処理し、単語ごとの特徴量を得ている。下段では画像から領域ごとの特徴量を抽出し、テキスト側の特徴と融合させることで、各領域がどの単語に対応するかのスコア行列を計算している。このように、単語と画像領域の対応関係を直接学習する構成になっていることが、GLIP の特徴である。

GLIP の大きな利点は、オープンボキャブラリ検出が可能である点にある。検出時には、従来のよ

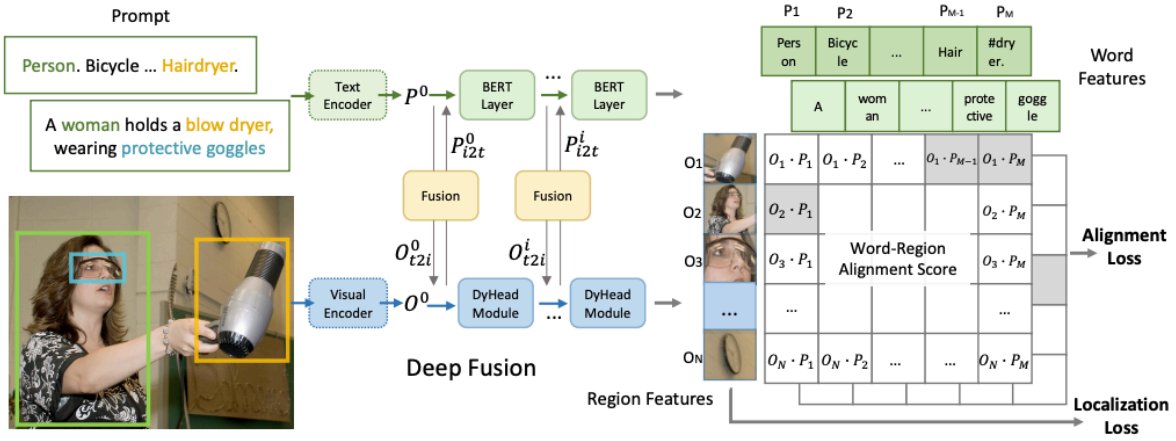

図 2.17　GLIP の全体アーキテクチャ [4]

うにあらかじめ固定されたクラス集合をモデル内部に持たせるのではなく、「girl」「smiling woman」のようなカテゴリ名や簡潔なフレーズをテキストとして入力し、そのテキストと画像中の領域の対応に基づいてバウンディングボックスとスコアを出力する枠組みとなっている。このため、クラス集合を事前に厳密に固定しておく必要がなく、タグ集合を後から追加したり、「女の子」「赤い服」など比較的柔軟な概念をタグとして扱える点は、タグ設計の自由度という観点から有用である。また、学習には物体検出データだけでなくキャプション付き画像も用いられているため、従来の検出器では学習データ数が少なく性能が出にくいカテゴリに対しても、ゼロショットもしくは少数ショットで一定の性能が得られやすいとされている。

一方で、GLIP を実際のシステムに組み込む際の課題も存在する。GLIP は大規模な Transformer を中核とするモデルであり、YOLO 系の軽量な検出器と比較すると、推論時間やメモリ使用量は大きい。また、GLIP は主として実写画像を用いたデータセットで事前学習されているため、線画や彩色済みセル画といったアニメーション画像とは分布が異なる。デフォルメされたキャラクタやアニメ特有の表現に対しては検出精度の低下が避けられないと考えられるが、アニメ領域ではバウンディングボックス付きの大規模教師データを新たに構築することは現実的ではない。そのため、本研究のような設定では、GLIP をアニメ画像に完全に適応させるための追加学習を前提とするのではなく、ゼロショットを前提としたある程度の誤差を許容した候補領域抽出として位置付ける必要がある。加えて、検出対象はテキストで指定するため、プロンプトとして与える語彙や表現の仕方によって検出結果が変動しやすく、アニメ制作におけるタグ設計に合わせて、どのような表現を用いるかといったプロンプト設計も重要な検討事項となる。

以上より、GLIP は、事前に固定されたクラス集合に依存しないオープンボキャブラリ物体検出を可能にする点で、アニメーションの修正カットに対して柔軟なタグ付けを行うための有力な候補である一方で、実写ベースの事前学習による精度低下やプロンプト依存性といった課題を抱えている。本研究においては、YOLO のようなクローズドセット検出器と比べてタグ語彙の自由度が高いことを重視しつつも、アニメ画像に対する検出精度や計算コストとのバランスを踏まえた上で、修正カットのタグ付けにおける適用可能性を検討する。

2.6.5 Grounding Dino [5]

Grounding DINO は Liu らによって提案されたオープンボキャブラリ物体検出モデルであり、DINO 系の高性能な Transformer ベース検出器と、GLIP のようなテキスト条件付きグラウンディングの枠組みを統合している [5]。すなわち、DINO が示した高精度な Transformer 検出アーキテクチャを土台としつつ、GLIP のようにテキストと画像の対応関係を明示的に学習することで、任意のテキストフレーズに対するゼロショット検出を実現したモデルと言える。図 2.18 に Grounding DINO の全体構成を示す。

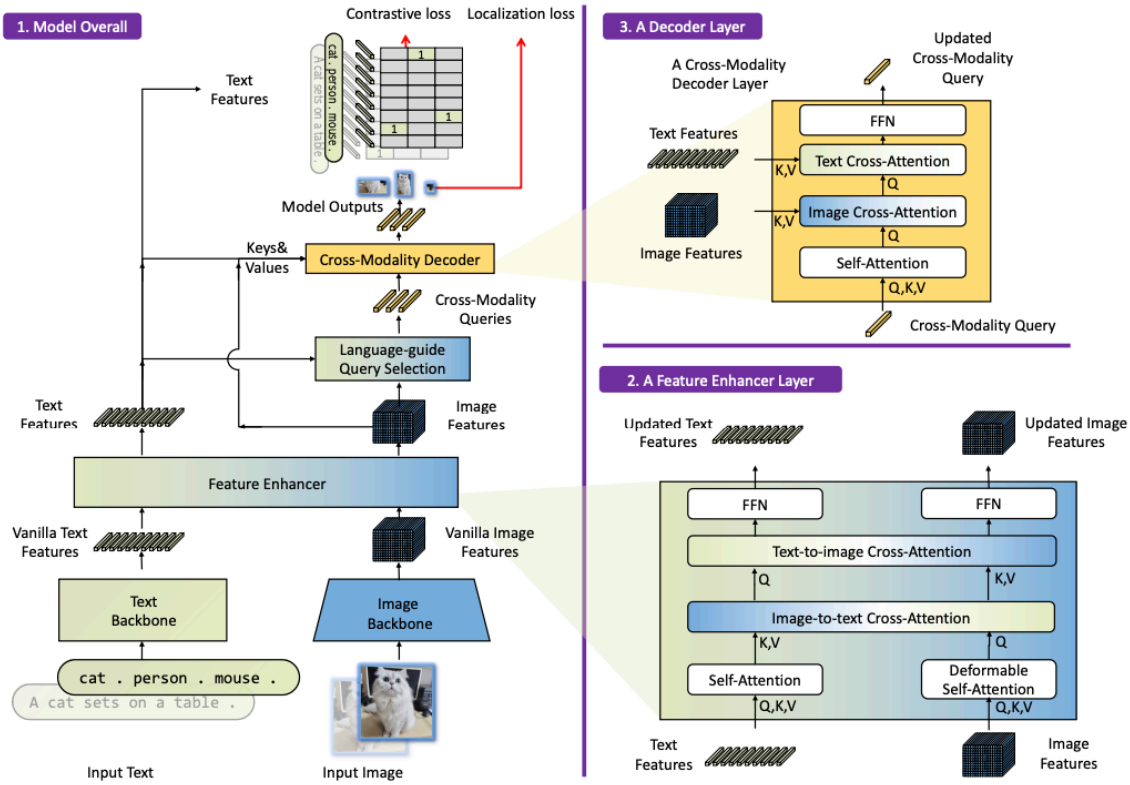

図 2.18　Grounding DINO の全体アーキテクチャ [5]

図の左下に示すように、入力テキストはテキストバックボーンを通して「cat」「person」「mouse」などのトークン列として埋め込みベクトルに変換される。一方で、入力画像は DINO と同様の画像バックボーンと FPN によってマルチスケールな画像特徴として表現される。これらの「素の」テキスト特徴と画像特徴は Feature Enhancer で統合され、テキストから画像へ、画像からテキストへという双方向のクロスアテンションを通じて、互いの情報を参照しながら更新された特徴表現へと変換される。

その上で、Language-guide Query Selection によってテキスト特徴から言語ガイド付きのクエリが生成される。これは DINO における学習済みオブジェクトクエリに相当するが、Grounding DINO ではテキスト側の情報を利用して「どのクエリがどのテキストトークンに対応するか」を明示的に制御する点が異なる。生成されたクエリは Cross-Modality Decoder に入力され、画像特徴に対するクロスアテンションとテキスト特徴に対するクロスアテンションを通して更新される。最終的な

デコーダの出力に対しては、テキストと画像領域の整合性を測るコントラスト損失と、バウンディングボックスの位置精度を評価するローカライゼーション損失が同時に適用され、テキストフレーズと画像中の領域が一貫した形で対応付けられるように学習される。

GLIP との比較という観点では、両者とも「テキストを入力として任意の語彙の物体検出を行う」という点で共通しているものの、GLIP が比較的シンプルな検出ヘッドの上にテキストとのクロスアテンションを載せた構成であるのに対し、Grounding DINO は DINO 系の強力な Transformer 検出アーキテクチャをそのまま活かしつつ、Feature Enhancer と Cross-Modality Decoder という二段構成で画像特徴とテキスト特徴を深く融合している点が大きな違いである。また、DINO と比較すると、DINO はあくまでクラス数が固定されたクローズドセット検出器であり、クラス埋め込みはモデル内部に固定されているのに対し、Grounding DINO はクラス名やフレーズをテキストとして与えることで、クラス集合を外から自由に指定できるオープンボキャブラリ検出器となっている。

利点としては、DINO 由来の高い検出精度と学習安定性を維持したまま、GLIP のように自然言語でカテゴリを指定できる柔軟なゼロショット検出を実現している点が挙げられる。特に高解像度画像に対する検出性能や、長いテキストプロンプトを扱うスケーラビリティに配慮した設計となっており、公開実装や事前学習済みモデルも整備されていることから、テキスト条件付き物体検出の実用的な選択肢として広く利用されている。

一方で、Grounding DINO は大規模な Transformer モデルを中核とするため、YOLO 系のような軽量な CNN ベース検出器に比べると推論時間やメモリ使用量が大きい。また、事前学習に用いられているデータセットの多くは実写画像であり、線画や彩色済みセル画といったアニメーション画像とは分布が異なる。そのため、本研究のようにアニメの修正カットに適用する場合には、デフォルメされたキャラクタやアニメ特有の背景表現に対する検出精度の低下をある程度許容した上で、ゼロショットな候補領域抽出やタグ候補生成のためのツールとして位置付ける必要がある。さらに、GLIP と同様に、検出対象はテキストプロンプトで指定するため、「どの表現でタグを与えるか」によって検出結果が変動しやすく、アニメ制作で用いるタグ語彙や修正指示の書き方に合わせたプロンプト設計も重要な検討事項となる。

このように Grounding DINO は、DINO の高精度 Transformer 検出器と GLIP 系の言語条件付きグラウンディングを統合したモデルとして位置付けられ、クラス集合を固定しない柔軟なタグ付けを可能にする一方で、計算コストやドメインギャップ、プロンプト依存性といった課題を抱えている。本研究では、YOLO や DINO のようなクローズドセット検出器と比較してタグ設計の自由度が高いという点を重視しつつ、アニメ画像に対する精度や処理コストとのバランスを踏まえて、修正カットのタグ付けにおける適用可能性を検討する。