マンガ会話コーパス構築における生成 AI 活用の可能性
―三つの OCR パイプラインと ChatGPT の比較―

王芳・金丸敏幸（京都大学）

Potential for Using Generative AI in Building a Manga Conversation Corpus: A Comparison of Three OCR Pipelines and ChatGPT

Wang Fang, Kanamaru Toshiyuki（Kyoto University）

要旨

本研究は、マンガ会話コーパス構築における生成 AI 活用の可能性を検討するため、同一 20 ページを対象に、mokuro＋manga-ocr、Manga Image Translator 48px、Koharu＋PaddleOCR-VL-1.6、ChatGPT を比較した。評価には領域対応 CER、文字領域 Recall・Precision、End-to-end 内容 CER、読み順ペア正解率、出力順全文 CER を用いた。全 157 領域では ChatGPT の領域対応 CER が 1.95%、End-to-end 内容 CER が 2.20%、文字領域 Recall が 99.36%で最良であった。記号・書き文字・擬音を除外した 125 領域では ChatGPT の領域対応 CER は 0.30%となった。以上の結果は、生成 AI が高精度な文字認識と構造理解を統合し、研究者自身によるマンガ会話コーパス構築の技術的負担を軽減しうることを示す。

１．はじめに

マンガは、言語情報と視覚情報が密接に結びついた複合的な媒体である。セリフや心内発話だけでなく、ナレーション、書き文字、擬音・擬態語、記号などが多様な方向・大きさ・字体で配置される。マンガの言語を量的に分析するためには、こうした文字情報を機械可読な形に転記し、読み順や話者等の情報を付与したコーパスが必要となる。しかし、その構築には文字領域の検出、文字認識、読み順の復元、話者・聞き手の同定など複数の工程を要する。

日本語マンガを大規模かつ体系的なアノテーション付きで研究利用できる公開資源は限られている。Manga109 は重要な研究基盤であるが、現代作品の収録には制約があり、市販マンガを用いた独自コーパスの公開・再配布にも著作権上の調整が必要となる。そのため、研究目的に応じて研究者自身が転記・注釈データを構築する場面が生じやすい。

一方、既存のマンガ処理ツールの多くは OCR、翻訳、閲覧支援等を主目的としており、会話文字列、読み順、話者・聞き手等を一貫して付与する「言語研究用コーパス構築」のための汎用ツールではない。複数のソフトウェアやモデルを組み合わせる作業は、特にプログラミングを専門としない人文学研究者にとって技術的な参入障壁となりうる。

近年、画像入力に対応する大規模マルチモーダルモデル（LMM）の発達により、文字認識とページ全体の構造理解を同一の対話的インターフェース上で行える可能性が生じている。そこで本研究では、同一のマンガ 20 ページを対象に三つの OCR パイプラインと ChatGPT を比較し、①文字認識精度、②文字領域の検出、③読み順、④記号・書き文字・擬音を除いた場合の精度変化を評価する。これにより、生成 AI がマンガ会話コーパス構築にどのように活用できるかを検証し、データ構築の負担軽減につながる可能性を考察する。

２．研究背景

２.１　公開マンガ資源とコーパス構築上の制約

マンガ研究におけるデータ基盤の整備は、著作権上の制約と不可分である。Aizawa et al. (2020) は、研究利用可能な高品質マンガデータセットが少ない主要因として、出版社・著者との著作権交渉および論文掲載時の許諾取得の困難さを指摘している。Manga109 はこの問題に対し、94 名の漫画家から許諾を得て 109 冊を研究用データセットとして整備したものであり、21,142 ページを含む。刊行年代の内訳は、1970 年代 7 冊、1980 年代 24 冊、1990 年代 45 冊、2000 年代 32 冊、2010 年代 1 冊である(Aizawa et al. 2020)。すなわち、Manga109 はマンガ画像処理研究の基盤として極めて重要である一方、2010 年代の作品が 1 冊にとどまるなど、現代のマンガ言語を直接反映する資料としては年代的制約を有する。

近年は Manga109 の注釈を再検討する Manga109-v2026 も提案され、約 29,000 件の対話注釈が修正されるなど、既存資源の高精度化が進んでいる(Baek et al. 2026)。しかし、これは Manga109 を基盤とする再注釈であり、研究対象となる作品群そのものを現代作品へ拡張するものではない。一方、研究者が市販の現代マンガを用いて独自コーパスを構築する場合、画像を含むデータの公開・再配布には権利処理が必要となるため、研究目的に合わせたコーパスを各研究者が研究内部で個別に構築せざるを得ない場面が多い。したがって、公開コーパスの拡充だけでなく、研究者自身が必要な作品から効率的に転記・注釈データを作成できる方法を確立することも、マンガ言語研究を発展させる上で重要である。

２.２　既存 OCR パイプラインとコーパス構築

一般的な文書 OCR と異なり、マンガでは縦書きと横書きが混在し、吹き出し外の書き文字が背景画像に重なり、字体や文字サイズも大きく変化する。本研究で比較した三つの専用系はいずれも、もともと「言語研究用コーパス構築」のみを目的として開発されたものではなく、マンガ閲覧、翻訳、ローカル編集等の用途に文字検出と OCR を組み込んだパイプラインである。以下、それぞれの目的と処理の流れを概観する。

mokuro は、マンガ画像をブラウザ上で選択可能なテキスト付きで閲覧することを主目的とするパイプラインであり、ページ画像から文字領域を検出し、検出領域を切り出した後、manga-ocr で日本語文字列を認識し、位置情報と認識結果を mokuro ファイルおよび HTML に保存する。Manga Image Translator（MIT）は、マンガ画像の文字検出、OCR、翻訳、画像処理を統合した翻訳パイプラインである。Koharu は、マンガ翻訳・編集を支援するローカルアプリケーションで、複数の検出器・解析器と OCR を組み合わせて処理できる。

会話コーパスの構築では、文字認識後に発話単位の整理、読み順の確定、話者・聞き手の同定、発話タイプの付与等が必要になる。また、文字領域を見落とせば発話そのものがコーパスから消失し、読み順を誤れば発話連鎖の復元が困難になる。そのため、既存 OCR を利用する場合でも、複数の処理結果を統合し、人手で確認・補完する工程が残る。さらに、擬音や記号をどの範囲までコーパス対象とするかによって見かけ上の OCR 精度は変化しうるため、文字認識、検出、順序を分離して評価し、研究目的に応じた対象範囲で評価する必要がある。

２.３　生成 AI を OCR として用いる意義

生成 AI をマンガ OCR に用いる最大の特徴は、文字認識と意味理解を同一の対話的インターフェース上で連続して行える点にある。専用 OCR では通常、画像から文字領域と文字列を抽出した後、話者や聞き手、発話タイプなどの注釈を別工程で付与する必要がある。こ

れに対し ChatGPT のような画像理解を備えた生成 AI では、ページ全体を見ながら文字列を抽出し、おおよその読み順を整理し、人物関係を推定して構造化した出力を生成できる。また、ローカル環境への複数モデルの導入、依存関係の調整、プログラムの実装を必ずしも必要とせず、対話形式で処理条件を指定できるため、プログラミングを専門としない研究者にも利用可能性が広がる。

一方、生成 AI には再現性の問題がある。専用 OCR は同一モデル・同一パラメータであれば比較的固定的な出力を期待できるのに対し、生成 AI はプロンプト、セッション、基盤モデルの更新等によって結果が変動しうる。このため、実用上の利便性と研究上の再現性を区別して評価する必要がある。

３．研究方法

３.１　対象資料と正解データ

対象は高松美咲『スキップとローファー』第 2 巻 pp.13–32 の連続 20 ページである。PDF から各ページを 200 dpi、1700×2200 ピクセルの JPEG 画像に変換し、四方式に同一画像を入力した。正解データは画像を目視して作成し、セリフ、心内発話、書き文字・擬音を領域単位で転記した。正解文字列が存在する 157 領域、空白除去後 1592 文字を評価対象とし、pp.13–32 の全 20 ページに読み順を付与した。

３.２　比較した四方式

表 1　比較方式と実験条件

方式	構成・モデル	処理形態
mokuro	mokuro 0.2.5＋manga-ocr 0.1.16	Google Colab
MIT 48px	Manga Image Translator（commit 95227a2…）＋ 48px OCR	Google Colab
Koharu	Koharu 0.61.2＋PaddleOCR-VL-1.6	Windows x64（headless API）
ChatGPT	GPT-5.6 Sol	Web UI

四方式の構成・モデルと処理環境は表 1 に示す。実験では、いずれも pp.13–32 の同一 20 ページを一括処理した。mokuro では manga-ocr の認識結果と文字領域を mokuro ファイルから取得した。MIT 48px では Local（Batch）モードで固定 commit、default detector（detection size=2048）、48px OCR を用い、翻訳と colorization は無効化した。Koharu では PP-DocLayout V3 による文字領域検出と PaddleOCR-VL-1.6 による文字認識を行い（Zhang et al. 2026）、ML Device は auto とした。

ChatGPT には GPT-5.6 Sol を使用し、pp.13–32 の 20 ページ画像を一度にアップロードした。単一実行で得られた出力を評価対象とし、入力したプロンプトは以下のとおりである。

マンガの OCR を行ってください。【条件】

1. セリフ、心内発話、ナレーション、書き文字をすべて抽出する。
2. 推測による修正や自然な日本語への言い換えを行わず、画像に見える表記をそのまま転記する。
3. 長音、促音、句読点、記号、伸ばし棒、三点リーダーを可能な限り保持する。

4. 読めない文字は勝手に補わず、［判読不能］と記載する。
5. ページ内の読み順に並べる。
6. 話者と聞き手情報及び話者と聞き手の性別情報を抽出してください。
7. 結果をエクセルに出力してください。

OCR 比較では文字列部分のみを用い、同時に得られた話者・聞き手等の付加情報は定量評価から除外した。

３.２.１　処理時間の記録

四方式の所要時間は、各方式で取得できた実行ログまたは実測値に基づいて記録した。pp.13–32 の 20 ページの処理時間は、mokuro＋manga-ocr が 304.54 秒（約 5 分 5 秒）、MIT 48px が 2,388 秒（約 39 分 48 秒）、Koharu＋PaddleOCR-VL-1.6 が 364.545 秒（約 6 分 5 秒）、ChatGPT が 507 秒（約 8 分 27 秒）であった。

mokuro＋manga-ocr、MIT 48px、Koharu＋PaddleOCR-VL-1.6 の三つの OCR パイプラインは、画像登録後の処理開始直前から全 20 ページの処理完了までを計測し、画像アップロード時間は含めなかった。ChatGPT は画像アップロード完了後から結果取得までの経過時間を記録した。

３.３　評価指標と再評価条件

評価では、Levenshtein (1966) の編集距離に基づく文字誤り率（Character Error Rate: CER）を用いた。CER は、正解文字列に対する置換（S）、削除（D）、挿入（I）の総数を正解文字数 N で除した値、すなわち CER=(S+D+I)/N とした。数値が小さいほど文字認識精度が高い。正規化では Unicode NFC を適用し、改行、半角・全角スペース、タブ等の空白のみを除去した。句読点、三点リーダー、長音、波ダッシュ、全角・半角記号の差は、全対象評価では原則として誤りとして数えた。

第一の指標は「領域対応 CER」である。正解データの文字領域と OCR 出力の文字領域をページ内で対応付け、同一領域と判断できた文字列だけを比較する。この指標は「検出できた領域を、どの程度正しく読めたか」を表す。ただし未検出領域を含まないため、単独ではシステム全体の性能を表さない。そこで第二に、文字領域の検出性能を評価するため、正解データに含まれる文字領域のうち OCR が検出できた割合を「文字領域 Recall」、OCR が検出した文字領域のうち正解領域と対応した割合を「文字領域 Precision」として算出した。第三に、文字認識だけでなく、文字領域の見落としや余分な検出も含めた総合的な文字誤り率を「End-to-end 内容 CER」として算出した。

第四に読み順を評価した。対応した OCR 領域の全ペアについて、正解読み順と前後関係が一致する割合を「読み順ペア正解率」とした。第五に、各ページの OCR 文字列をツールの出力順のまま連結し、正解文字列を正しい読み順で連結して比較する「出力順全文 CER」を算出した。これは文字認識、検出漏れ、余分検出、読み順のずれが最終的な連続テキストに与える影響をまとめて示す指標である。

さらに、マンガでは、会話本文と比べて書き文字、擬音、単独記号の視覚的変異が大きい。そこで、会話コーパスの主要対象をセリフ・心内発話の言語内容と想定した補助分析を行った。まず、正解データで「書き文字・擬音」と分類した 27 領域を除外した。次に、残るセリフ・心内発話のうち文字ではなく記号だけから成る 5 領域を除外した。さらに、残った領域については Unicode の句読点・記号類を評価対象から除外し、文字内容に焦点を当てて再

評価した。この再評価では 125 領域・1344 文字を対象とした。

４．結果

４.１　全 157 領域を対象とした評価

表 2　全対象における四方式の評価結果

方式	領域 CER	E2E CER	領域 Recall	領域 Precision	読み順	全文 CER
Koharu	2.77%	7.47%	89.17%	99.29%	97.29%	15.58%
MIT 48px	4.22%	10.24%	80.89%	**100.00%**	96.60%	17.59%
mokuro	9.79%	14.45%	86.62%	99.26%	91.05%	32.79%
ChatGPT	**1.95%**	**2.20%**	**99.36%**	99.32%	**98.01%**	**8.67%**

表 2 に示すように、全対象評価では ChatGPT が領域対応 CER 1.95%で最も低く、対応できた文字領域の認識精度が最も高かった。Koharu は 2.77%でこれに続き、MIT 48px は 4.22%、mokuro は 9.79%であった。検出漏れと余分検出を含む End-to-end 内容 CER でも ChatGPT が 2.20%で最小であり、Koharu 7.47%、MIT 10.24%、mokuro 14.45%との差が大きかった。

検出性能を見ると、ChatGPT の文字領域 Recall は 99.36%で、157 正解領域のうち 156 領域に対応した。Koharu は 89.17%、mokuro は 86.62%、MIT は 80.89%であり、従来型パイプラインでは文字領域の見落としが全体誤差を押し上げていた。一方、文字領域 Precision はいずれも 99%以上で、MIT は 100.00%であった。このことは、本データでは「存在しない文字を誤検出する」ことより、「存在する領域を取りこぼす」ことの方が主な問題であったことを示す。

読み順ペア正解率は ChatGPT が 98.01%、Koharu が 97.29%、MIT が 96.60%、mokuro が 91.05%であった。さらに出力順全文 CER は ChatGPT 8.67%、Koharu 15.58%、MIT 17.59%、mokuro 32.79%である。領域対応 CER より全文 CER が大幅に高いことから、個々の文字を正しく読めるかだけでなく、どの領域を拾うか、どの順に並べるかがコーパス化の品質を大きく左右していることが分かる。特に mokuro では領域認識そのものの誤りに加えて、出力順のずれが連続テキスト化に影響している。

４.２　記号・書き文字・擬音を除外した評価

表 3　記号・書き文字・擬音を除外した評価結果（125 領域・1344 文字）

方式	領域 CER	E2E CER	領域 Recall	領域 Precision	読み順	全文 CER
Koharu	0.82%	0.97%	**100.00%**	99.21%	97.49%	8.41%
MIT 48px	1.43%	2.23%	94.40%	**100.00%**	96.92%	9.67%
mokuro	0.75%	1.19%	99.20%	99.19%	91.15%	22.77%
ChatGPT	**0.30%**	**0.45%**	**100.00%**	99.13%	**99.06%**	**4.76%**

除外後の結果は、全対象評価とは異なる特徴を示した。ChatGPT の領域対応 CER は 0.30%、

End-to-end 内容 CER は 0.45%まで低下し、依然として最良であった。Koharu は領域対応 CER 0.82%、End-to-end 内容 CER 0.97%、mokuro はそれぞれ 0.75%、1.19%となり、両者の文字認識精度は非常に高い水準に近づいた。MIT も領域対応 CER 1.43%、End-to-end 内容 CER 2.23%まで改善した。

とりわけ mokuro の変化が大きい。全対象の領域対応 CER 9.79%から除外後 0.75%へ約 9.04 ポイント低下し、End-to-end 内容 CER も 14.45%から 1.19%へ約 13.26 ポイント低下した。Koharu の文字領域 Recall は 89.17%から 100.00%、mokuro は 86.62%から 99.20%、MIT は 80.89%から 94.40%へ上昇した。したがって、従来型 OCR パイプラインの検出漏れ・認識誤りの相当部分が、書き文字・擬音・単独記号および記号表記に集中していたと考えられる。

なお、除外後の文字領域 Precision は Koharu 99.21%、MIT 100.00%、mokuro 99.19%、ChatGPT 99.13%であり、依然として高い。したがって、全対象での性能差を「誤検出の多さ」だけで説明することはできない。むしろ、マンガ特有の視覚表現を正解領域としてどこまで含めるか、その領域を検出できるか、記号をどのように転記するかという評価設計が重要である。

５．考察

５.１　ChatGPT と従来型 OCR の性能特性

全対象・除外後のいずれにおいても、ChatGPT は領域対応 CER と End-to-end 内容 CER が最も低く、文字領域 Recall も高かった。特に全対象で高い Recall を示したことは、ページ全体の視覚文脈を利用して、吹き出しや小さな文字領域を広く拾える可能性を示している。これは、文字検出器と文字認識器を明示的に分離する従来パイプラインとは異なる強みである。

ただし、「ChatGPT が OCR 一般で常に専用モデルより優れる」と結論づけることはできない。Shi et al. (2023) が示すように、LMM の OCR 能力は言語やタスクによってばらつきがあり、専用 OCR の研究価値は依然大きい。本研究は一作品・20 ページという限定されたサンプルであり、日常系マンガの比較的整った会話文字が中心である。異なる作家、少年マンガ、ファンタジー、手書き文字の多い作品、低解像度画像等では順位が変化する可能性がある。

一方、除外後評価で最も注目すべき点は、mokuro の領域対応 CER が 0.75%まで低下し、Koharu の 0.82%をわずかに上回ったことである。これは、mokuro＋manga-ocr が通常のセリフや心内発話の文字認識そのものに弱いわけではなく、全対象で高かった CER の主因がマンガ特有の書き文字・擬音・記号表記にあったことを示す。manga-ocr は吹き出し単位の複数行文字を一度に認識できるよう設計されており、会話コーパスの中心となる整った日本語文字列に限定すれば十分高い性能を持つと評価できる。

この結果はコーパス設計上も重要である。研究目的が「セリフを中心とした会話コーパス」であり、擬音や視覚効果文字を分析対象としないのであれば、OCR 評価時にそれらを別層として扱う方が実用的である。逆に、マンガ表現論やオノマトペ研究では、書き文字・擬音を除外すること自体が研究対象を損なうため、全対象評価を用いる必要がある。すなわち、OCR の精度は単一の数値ではなく、「何をコーパスの文字と定義するか」に依存する。

５.２　読み順・構造化情報と実用的なコーパス構築

会話コーパスの利用では、正確な文字列だけでなく発話の順序が不可欠である。除外後、

mokuro は領域対応 CER が低かった一方、出力順全文 CER は依然として高く、個々の文字認識が高精度でも、出力順が正解の読み順とずれると会話データとして利用しにくいことが分かる。これに対し ChatGPT は読み順ペア正解率が最も高く、出力順全文 CER も最も低かったことから、ページ全体の構造理解が連続テキスト化に寄与したと考えられる。

さらに、ChatGPT では OCR と同時に話者、聞き手、話者・聞き手の性別等を表形式で出力させることができた。これらは本稿では精度評価していないため「正しい」とは断定できないが、コーパス構築工程を考えると重要な可能性である。専用 OCR では「画像→文字列」の後に人手または別モデルで話者情報を付加するのに対し、生成 AI では「画像→文字列＋談話情報」の一体処理が可能である。将来的に話者同定の正解データを整備し、精度を定量化できれば、生成 AI の有効性を OCR 精度より広いコーパス構築コストの観点から評価できる。

表 4　マンガ会話コーパス構築における各方式の位置づけ

観点	mokuro	MIT 48px	Koharu	ChatGPT
会話文の認識精度	高い	高い	高い	最も高い
書き文字・擬音・記号への対応	誤りが増加	誤り・漏れが増加	漏れが増加	相対的に強い
読み順精度	要確認	比較的高い	高い	最も高い
話者・聞き手付与	別工程	別工程	別工程	同時出力可能
再現性	高い	高い	高い	出力変動あり

いずれの方式でも、最終的な研究データとして用いる際には人手による確認が必要である。特に ChatGPT が出力した話者・聞き手情報は本稿で定量評価していないため、その利用には追加の検証を要する。

５.３　本研究の限界と今後の課題

本研究には少なくとも五つの限界がある。第一に、対象が一作品 20 ページに限定されており、作品ジャンル、作画スタイル、印刷品質の差を一般化できない。第二に、ChatGPT はサービス側で継続的に更新されるため、将来同一プロンプトを用いても完全に同じ出力が得られるとは限らない。第三に、生成 AI の確率的な出力揺れを測るための反復試行を行っていない。第四に、所要時間は各方式について 1 回の測定値であり、反復測定による処理時間の変動幅を検証していない。第五に、話者・聞き手・性別など、会話コーパス構築にとって重要な意味情報については、今回定量的な正解率を測定していない。

また、正解データの設計自体が評価値に影響する。「〜」と「～」のような表記差、単独記号、書き文字を正解としてどのように扱うかによって CER は変化しうる。Baek et al. (2026) が Manga109 の再注釈で指摘するように、マンガ OCR の評価には認識モデルだけでなく、アノテーション基準の精緻化が不可欠である。今後は、セリフ層、書き文字・擬音層、記号層を分けた多層正解データを作成し、目的別に精度を提示することが望ましい。

６．おわりに

本研究は、高松美咲『スキップとローファー』第 2 巻 pp.13–32 の 20 ページを対象に、三

つの OCR パイプラインと ChatGPT を比較し、マンガ会話コーパス構築における生成 AI 活用の可能性を検討した。全 157 領域の評価では、ChatGPT が領域対応 CER、End-to-end 内容 CER、文字領域 Recall、読み順ペア正解率、出力順全文 CER で総合的に最も良好な結果を示した。

書き文字・擬音および記号を除外した 125 領域では、ChatGPT に加えて mokuro、Koharu、MIT も大幅に改善した。従来型 OCR の主要な誤りが通常の会話文よりもマンガ特有の視覚表記に集中していたことから、会話コーパスを構築する際には、研究目的に応じてセリフ、書き文字・擬音、記号を分けて評価する必要がある。

以上より、生成 AI は文字認識に加えて読み順や話者・聞き手等の談話情報を同一画像から構造化できる点で、マンガ会話コーパスの初期データ作成に活用できる可能性が示された。一方で、本研究は一作品 20 ページの単一実行に基づくため、一般化と再現性には限界がある。今後は対象作品・ページ数を拡大し、複数回実行による再現性、処理時間・費用、話者同定精度、人手修正時間を含めた総合評価を行う必要がある。

文　献

Kiyoharu Aizawa, Azuma Fujimoto, Atsushi Otsubo, Toru Ogawa, Yusuke Matsui, Koki Tsubota, Hikaru Ikuta (2020). “Building a Manga Dataset ‘Manga109’ With Annotations for Multimedia Applications”, *IEEE MultiMedia*, 27:2, pp. 8–18. https://doi.org/10.1109/MMUL.2020.2987895

Jeonghun Baek, Atsuyuki Miyai, Shota Onohara, Hikaru Ikuta, Kiyoharu Aizawa (2026). “Manga109-v2026: Revisiting Manga109 Annotations for Modern Manga Understanding”, *Culture × AI Workshop at ICML 2026*. https://arxiv.org/abs/2605.21182

Vladimir I. Levenshtein (1966). “Binary Codes Capable of Correcting Deletions, Insertions, and Reversals”, *Soviet Physics Doklady*, 10:8, pp. 707–710.

Yongxin Shi, Dezhi Peng, Wenhui Liao, Zening Lin, Xinhong Chen, Chongyu Liu, Yuyi Zhang, Lianwen Jin (2023). “Exploring OCR Capabilities of GPT-4V(ision): A Quantitative and In-depth Evaluation”, arXiv:2310.16809. https://arxiv.org/abs/2310.16809

Zelun Zhang, Hongen Liu, Suyin Liang, Yubo Zhang, Yiqing Xiang, Jiaxuan Liu, Ting Sun, Manhui Lin, Yue Zhang, Changda Zhou, Tingquan Gao, Cheng Cui, Yi Liu, Dianhai Yu, Yanjun Ma (2026). “PaddleOCR-VL-1.6: Expanding the Frontier of Document Parsing with Under-Optimized Region Refinement and Progressive Post-Training”, arXiv:2606.03264. https://arxiv.org/abs/2606.03264

関連 URL

kha-white. manga-ocr (v0.1.16). GitHub repository. https://github.com/kha-white/manga-ocr （2026 年 8 月 9 日閲覧）.

kha-white. mokuro (v0.2.5). GitHub repository. https://github.com/kha-white/mokuro （2026 年 8 月 9 日閲覧）.

zyddnys. manga-image-translator (commit 95227a2bb0fd306cd4f0c104d57284026f991b3a). GitHub repository. https://github.com/zyddnys/manga-image-translator/commit/95227a2bb0fd306cd4f0c104d57284026f991b3a （2026 年 8 月 9 日閲覧）.

mayocream. Koharu (v0.61.2，本実験で使用したアプリケーション版). GitHub repository. https://github.com/mayocream/koharu （2026 年 8 月 9 日閲覧）.

PaddleOCR. “PaddleOCR-VL-1.6 Introduction”. PaddleOCR Documentation. https://www.paddleocr.ai/main/en/version3.x/algorithm/PaddleOCR-VL/PaddleOCR-VL-1.6.html （2026 年 8 月 9 日閲覧）.
OpenAI. “ChatGPT Image Inputs FAQ”. OpenAI Help Center. https://help.openai.com/en/articles/8400551-image-inputs-for-chatgpt-faq （2026 年 8 月 9 日閲覧）.