令和 6 年度 卒業論文

顔識別モデルと生成 AI を用いた

動画内容の解析

2025 年 2 月 13 日

静岡大学 工学部 数理システム工学科

岡部研究室

5011-6034 角田梓美

目次

第1章　序論

現在,Tver や Netflix のような動画配信サービスにおいて,高品質なコンテンツが増加している.例えば,視聴履歴からおすすめの作品を表示する機能や,映像に関する説明をナレーションによって補足する「解説放送版」の配信などが挙げられる.このようなコンテンツの増加に伴い,動画配信サービスの需要がさらに高まっている.そこで本研究では,配信動画の視聴体験の充実を目指し,映像やテキストデータから映像の登場人物やシーンの内容を解析するシステムを開発した.

一方で,ドラマなどの映像作品を見ない理由として「登場人物が多く,長くて複雑なストーリーでは,人物同士の関係性や内容を忘れてしまい,物語を追いづらくなる」という声も見られる.ドラマを見るのが好きな自分としては,そのような理由でよい作品との出会いを避けてしまっているのではないかと寂しい気持ちにもなることがある.しかし実際自分も,ドラマを視聴していると「この人は何者か？」,「主人公とはどのような関係だったか？」と思い出せなくなることがある.特に登場人物が多い作品や複雑なストーリーの作品では,一度わからなくなってしまうとそこから物語を追うのが難しくなることは少なくない.

そこで本研究では,そういった悩みを改善し,配信動画の視聴をより豊かなものにするために,映像やテキストデータから映像の登場人物やシーンの内容を解析するシステムを開発した.話者やセリフ,その時点の秒数を記録したテキストファイルと,顔写真と人物名を紐づけたデータさえあれば,オープンソースライブラリである InsightFace による高精度の顔検出,顔識別と GPT-4 による解析力を活用し,指定したシーンまでのストーリーや人物関係を知ることが可能となる.

先ほど例に挙げた Tver や Netflix にも,配信動画のあらすじや出演者情報を表示する機能がある.また,公式サイト等にアクセスすれば,それ以前のあらすじや相関図なども見ることができる.ただ,これらは静的な情報にとどまり,視聴中の動画に連動した情報提供は行われない.要するに,動画の視聴を一時停止し,その間にあらすじやキャラクター情報を確認することはできるが,動画の進行を考慮したリアルタイムな情報を得ることは困難である.一方このシステムを導入すれば,動画の進行に合わせた情報が表示さるため,視聴中に必要な情報を即座に得ることができ,物語への没入感も向上するだろう.

第2章　関連研究

2.1　Anim-Director

GPT-4 や MidJourney[1],Pika[2]などの生成 AI を活用し,ユーザーが入力した文章からアニメーションを自動的に生成するという研究がある.Anim-Director[3]は,「文章の補足」,「スクリプト生成」,「シーン画像の生成」,「画像の改善」,「動画生成」,「動画の改善」という 6 つのステップを経て高品質な動画を生成している.

まず「文章の補足」では,入力した簡潔な文章を GPT-4 に補足してもらう.このステップでは,物語の構造の不足部分や会話,背景描写が追加される.次の「スクリプト生成」では,GPT-4 によって主要な登場人物や細かい人物設定を補足し,それを元にシーン別の細かい説明を生成する.これは,画像や動画を生成するための設計図となる.

その文章を基に,「シーン画像の生成」では,MidJourney を用いてアニメーションの基盤となる画像を生成する.まず,キャラクターと背景が生成され,アニメーション全体の一貫性を確保する.続いて,各シーンの画像を生成する.この際,GPT-4 が各シーンに適したプロンプトを生成し,それを MidJourney に送信することでシーンごとの初期画像を取得する.ここで,AI による画像生成のランダム性を考慮し,次に「画像の改善」を行う.生成された画像を,一貫性や文章との整合性に基づき GPT-4 が評価し,結果によって画像を修正しながら基準を満たすまで検証を繰り返す.この時,画像の修正にはセグメンテーションモデルである SAM[4]が用いられる.

この検証が終了すると,「動画生成」のステップに入る.このプロセスでは Pika という動画生成ツールを使用する.「シーン画像の生成」で得られた画像と,キャラクターの動きや感情に基づき GPT-4 が生成したプロンプトを Pika へ入力すると,動画が生成される.最後に,「動画の改善」を行う.Pika は,動画候補として 10 種類の動画を出力する.この中から,文章とのブレや背景とオブジェクトの一貫性を指標とした評価を行い,候補を 3 つに絞る.その後,GPT-4 を用いてアニメーションの動作や文章との整合性を確認する.そこで 3 つの候補から絞られた 1 つの動画が,最終結果となる.

このように Anim-Director は,複数の生成 AI を組み合わせることで,ユーザーからの簡単な指示のみで一貫したアニメーションを自律的に生成できる画期的なシステムである.

2.2　顔識別モデル

顔識別モデルは，深層学習を用いて個人の顔の特徴を抽出し，顔検出や識別を行う技術である．代表的なモデルには,「顔の数値ベクトル化」を確立させ AI 顔認識の歴史に大きく影響を与えた FaceNet[5]や，リアルタイムの認識に強みを持つ DeepFace[6]，余弦類似性の基づいた距離で識別する CosFace[7], 角度ベースで分類を行う SphereFace[8]などがある．それぞれ異なるアプローチで性能を向上させており，応用範囲も広がっている．

第 3 章　既存手法

3.1　InsightFace

特定のシーンに写る出演者を識別するため,オープンソースの顔解析ツールボックスである InsightFace[9]を導入している.InsightFace は,画像中から顔の領域を抽出する「顔検出[10]」,顔領域から特徴量(埋め込みベクトル)を抽出する「顔特徴量抽出」,特徴量から人物を特定する「顔認識」という三つの役割を担う.その中核となる技術として,ニューラルネットワークを利用し学習する深層学習や ArcFace 損失関数が用いられる.

3.1.1 深層学習ベースのアーキテクチャ

InsightFace は,ResNet[11]などのニューラルネットワークをベースにした深層学習モデルを採用している.

一般的に,深層学習モデルは層が深くなるにつれより高度で複雑な特徴を抽出でき,性能が向上すると言われている.その一方で多層化は課題もあり,層が深くなることで学習時誤差が伝播されず,勾配という誤差の調整に重要な要素がゼロに近づき学習が進まなくなるという「勾配消失問題」に直面する.この問題を解決するために提案されたのが ResNet であり,152 層という深いネットワークを持ちながら,性能を落とさず学習をすることが可能となる.

これには,勾配消失問題を解決するため,「残差学習」が取り入れられている.残差学習は層が深くなっても効率的に学習を進められるようにする手法であり,従来のように新しい出力を一から計算するのではなく,層の出力に前の層の出力を直接加算し,入力との変化のみを学習する考え方である.これにより,情報が失われにくく学習がよりシンプルになり,深いネットワークでも性能が低下しにくくなる.

3.1.2 ArcFace 損失関数

ArcFace[12]は,顔認識において高精度を実現するための損失関数で,特徴ベクトルの識別性を高める方法を提供している.ニューラルネットワークは顔画像から特徴を抽出し,それを数値の並びである特徴ベクトルとして表現する.

まず,ある顔写真から抽出された特徴ベクトルと,特定のクラス(人物の顔)を表現しているクラス代表ベクトルの二つを正規化することで,比較基準をベクトル間の角度に統一する.これにより,ベクトルの規模に依存しない分類が可能となる.

次に,逆余弦関数を用い,二つのベクトルの角度を計算する.その角度差が,二つのベクトル

がどれだけ似ているかを示す値となる.その際,同じクラスの特徴ベクトルとクラス代表ベクトルの間に加算的角度マージン(additive angular margin)を加え,分類の基準をより厳しくする.例えば,角度差 30° 以内で同じクラスと判定されるとすると,通常であれば特徴ベクトルとクラス代表ベクトルの角度差が30° 以内であれば同一人物と判定される.ここで加算的角度マージンを 10° 追加すると,元の角度に＋10° して分類されるため,実際の角度差が30° であると同じクラスとは認定されない.このマージンの追加により,同じ人物の特徴ベクトルはより近く,異なる人物の特徴ベクトルはより遠くに配置されるよう学習される.

こうして正解クラスに対し厳しい条件を課し,識別性能を向上させている.

3.2 生成 AI

シーンの内容や登場人物の解析,結果の言語化の役割は,GPT-4[13]が担っている.GPT-4のような大規模言語処理モデル(LLMs)の開発は,自然言語処理の急速な進歩に繋がっている.LLM は大規模なデータセットによりトレーニングされており,さらに文字だけのテキストデータを提供すれば自動的に学習するという能力も持つ.GPT-4 は Transformer という言語モデルを基盤としており,今日ではテキストデータだけでなく画像や動画など,視覚的データに対するタスクにも対応している.

3.2.1 GPT-4 の仕組み

GPT-4 の仕組みをいくつかのステップにわけて説明する.

-1.前処理とトークン化

ユーザーが入力したテキストから,不要なスペースの削除や特定の文字の正規化が行われ,モデルが理解しやすい形式に変換される.そこから,文章をトークンに分割する.例えば「AIの歴史をおしえてください」という文章であれば,”AI”,”の”,”歴史”,”を”,”教えて”,”ください”というように,テキストを構成する最小単位に分割する.

-2.エンコーディングと埋め込み

各トークンがそれぞれに意味を持たせた数値ベクトルに変換され,その後文脈や関係性を考慮した埋め込みベクトルに変換される.ここで生成される埋め込みベクトルは単語の意味を持っており,意味が似ている語同士は近い位置に配置される.

また,意味だけでなく文中のトークンの位置も記録し,順序や依存関係を理解する必要がある.モデルは,この埋め込みベクトルに位置的エンコーディングを追加し,「AI」は文頭,「歴史」は文中,などというように各トークンがどこにあるかを記録する.

-3. トランスフォーマーによる処理

プロセスの核心部分となるトランスフォーマーの中心的な機構である「自己注意機構」が

トークン間の関係性を計算する.すべてのトークンの埋め込みベクトルを,必要とする情報を問い合わせるための「クエリ」,どのような情報を持つかを表現する「キー」,実際の情報を保持する「バリュー」という三つのベクトルに変換する.変換後,クエリとキーの類似度に基づき重みを計算し,最終的にバリューを加重平均して結果となるデータを生成する.ここでの類似度とは,「このトークン同士はどれくらいどれくらい関係があるか(注意度)」を示すものである.

-4.後処理

トランスフォーマーにより生成されたトークン列を,自然な言語に近づけるための調整が行われる.文法や不自然な表現の修正が行われた後,結果としてそのテキストが提供される.

このようなステップを通し,自然で適切な応答を生成している.さらに GPT-4 では,ファインチューニングや人間によるフィードバックを活用した強化学習が行われており,日々モデルが改善されている.

第 4 章　提案手法

本論文では,InsightFace と GPT-4 を活用し,動画のストーリー理解を補助するシステムを提案する.システムの概要を図 1 に示す.「シーン内の顔識別」,「ストーリー要約」,「関連シーン検索」という三つの主要なステップを通じ,ユーザーの動画視聴をサポートする.

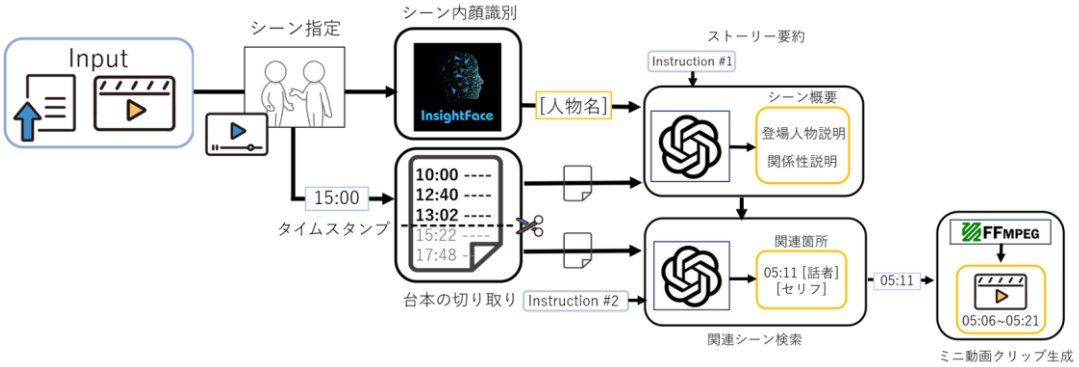

図 1 システムの流れ

4.1 ユーザーインターフェース

Web アプリでは,まずあらかじめ用意してある解析対象となる動画とその台本を,図 2 内の①,②のボタンよりそれぞれアップロードする.

図 2 アップロードボタン

そうすると,図 3 のように動画再生プレイヤーが表示される.動画の再生中,映っている人物の解析をしたい時点でまず,図 3 の「Recognize Faces」(③)を押す.すると,⑤ようにフレームに映る人物が表示される.その後,「Analyze」(④)のボタンを押下すると,ボックスに解析時点までの人物の説明や関係性が表示される(⑥).

図 3 解析ボタン(上),結果の表示(下)

※顔解析のため登録した顔写真と人物名を紐づけたデータから,写真と名前を取得し表示する.

⑥に表示された文章は一文ごとに選択できるようになっており,特定の出来事の関連シーン位置を知りたい場合に活用できる.例えば,⑥の Character 内の「自分とは正反対の B に興味を持ち,入れ替わりという想像を軽やかに受け入れる柔軟さを持つ」という文章をクリックすると,⑦のように関連するセリフが表示される.更にタイムスタンプ部分(⑧)をクリックすると,短い動画クリップがポップアップ形式で表示される.

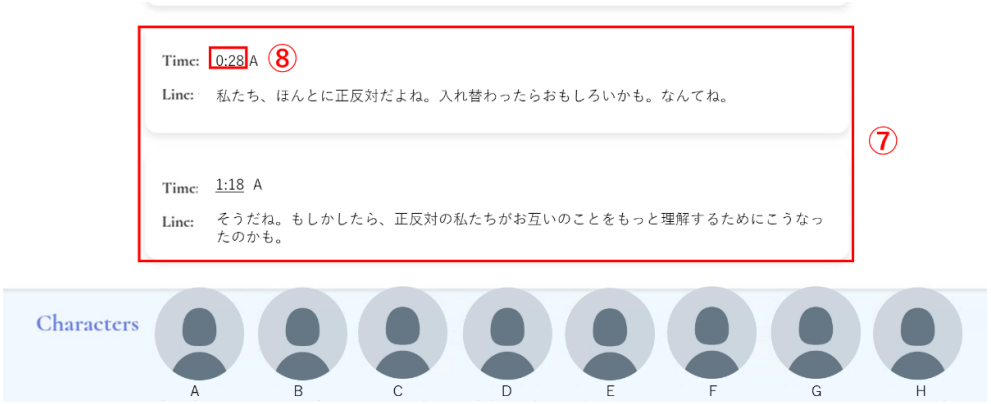

図 4 関連シーン検索の結果表示

4.2 事前準備

動画のデータセットへの事前学習は不要だが,解析対象の動画,タイムスタンプや話者が記載された動画の台本,出演者ごとの名前と顔写真データを事前に準備しておく必要がある.

台本の作成には,「CLOVA Note[14]」という AI 音声認識アプリケーションを用いている.このアプリケーションは,音声ファイルをアップロードするだけで話者を判別し,図 5(左)のようなテキストを生成する.これに図 5(右)のように重要なシーンの遷移や情景に関する情報を加えたものを,システムに入力する台本として使用する.

InsightFace による登場人物の特定に利用する顔写真データについては,登場人物一人に対し登録できる枚数に制限はない.ただし,特に登場シーンが多い人物については,さまざまな角度から撮影された写真を複数登録しておくことをお勧めする.これにより,より正確な認識が可能となる.

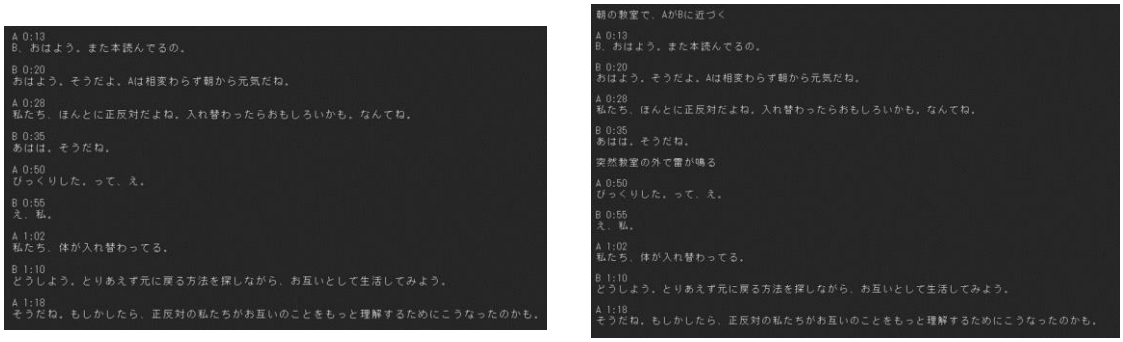

図 5 動画の台本例 情報付加前(左),情報付加後(右)

4.3 InsightFace を用いたシーン内顔識別

動画の再生中,ビデオプレイヤーの下にある「Recognize Faces」のボタンを押下すると,その時点のフレームに映る人物名を InsightFace により特定し,即時にサイト上に表示する.事

前に登録した顔写真の埋め込みベクトル間の類似度を計算し,類似度がしきい値を超える場合対応する人物名を表示するが,システムの認識精度に応じてしきい値の設定を調節することも可能である.

4.4　GPT-4 による人物・ストーリー解析

フレームに映る人物名が表示された後,「Analyze」のボタンを押下すると,GPT-4 によってその時点までの内容の解析が始まる.まず,ドラマの視聴中にリアルタイムでシステムを利用することを考慮し,物語のネタバレを防ぐために台本の編集を行う.台本内のセリフに与えられたタイムスタンプを用い,「Recognize Faces」のボタンを押下した時点以降の内容を全て削除した台本を生成し,新たなテキストファイルとして保存する.

次に,GPT-4 に新しい台本とプロンプトを送信し,結果を取得する.この際,GPT-4 に送信するプロンプトで,表 1 のように回答形式を指定する.それにより json 形式の回答が得られるため,追加で変換処理を行う必要がなく,JavaScript で直接処理し,適切にサイト上に結果が表示できる.

4.4.1 ストーリー要約のプロセス

このシステムでは,ストーリー要約の生成のプロセスにおいて,ユーザーが求める回答に近づけるため,二段階で解析をする手法を採用した.

まず,解析時点までに発生した物語の「重要な出来事」を,GPT-4 を用いて抽出する.そしてその出来事(コード内では“response1_content”)を踏まえ,表 1(Instruction #1.2)のようなプロンプトで人物や関係性についての分析をするよう GPT-4 に要求する.

この結果,一段階目で人物や関係性に関する分析の際の基盤を提供することができる.もしこの手法を用いない場合,「物語の中心人物で,明るい高校生」というような,当てはまる人が多そうな一般的で無難な回答を返す場合が多くある.一方,二段階解析を用いると,「明るくて優しい高校生だが,……をきっかけに,……となり,生活が一変する」というように,人物像と物語の流れを適切に把握できる回答を得ることができる.これは,単なる人物紹介にとどまらず物語の進行や登場人物に深く関わる情報を得られる点で有用であることがわかる.

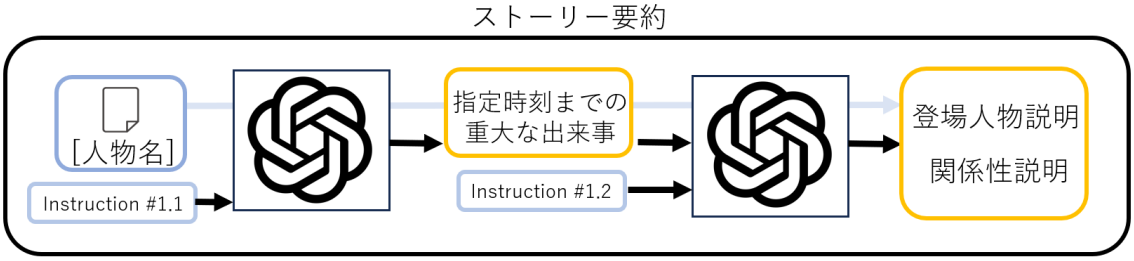

図 6 ストーリ要約のプロセス

4.5 関数シーン検索

本システムでは,解析結果や視聴者の選択に基づき関連シーンを検索できる機能を備えている.4.3.1 で生成された文章が一文ごとに選択できるようになっており,視聴者が特定の文をクリックすると,表 1 のプロンプト(Instruction #2)とともに文章(“selected_sentence”)が GPT-4 に送信される.そうすると,図 1 の台本の切り取りのプロセスで切り取られた台本の内容を GPT-4 が分析し,選択された説明文の根拠となるようなセリフとタイムスタンプを特定する.この際,関連性の高い 2 つのシーンを抽出し,それらを json 形式でシステムに返却し,サイト上に表示する.

さらに,特定されたタイムスタンプを基に,動画の変換,切り出し,再生などができるソフトウェアである FFmpeg[15,16]を用い,該当シーン周辺 15 秒間を切り取った動画クリップを生成する.これは,視聴者がサイト上で簡単に再生できるようになっている.

Instruction #1.1

現在までに起こった重大な出来事を,ひとつ 50 字以内で時系列順に箇条書きで{num}つだけあげてください

Instruction #1.2

{response1_content} の出来事に焦点を当てて

A. 認識された人物について，性格や物語の重要な要素を 150 文字で説明

B. 認識された人物間の関係性や重要エピソードを簡潔に 250 文字で説明. 認識人物が一人の場合は，主要な周辺人物との関係性を解析

台本からわからないことは言及しないでください

1. 認識人物: {', '.join(recognized_names)}

2. 解析時点 {formatted_time}

【回答形式】:

```
{{
  "role_and_character": {{
    "人物名 1": "説明",
    "人物名 2(いれば)": "説明"
  }},"
  "relationship": "説明"
 }}
```

Instruction #2

台本から,「{selected_sentence}」という説明や内容に関連する根拠となる部分を関連度順に 2 つ特定し,該当箇所のタイムスタンプ,セリフ,そのセリフの話者を抽出してください

結果を以下の JSON 形式で返してください:

```
{{
"scenes": [
    {{
    "timestamp": "タイムスタンプ",
    "speaker": "話者の名前",
    "text": "セリフ"
    }}
]
}}
```

表 1 GPT-4 に送信するプロンプト

第 5 章　結果と考察

5.1　InsightFace の精度

まずこのシステムで重要となるのが,顔認識の精度である.ここでは,高難易度な顔認識の精度を調べるため,実際の動画のシーンから抽出した, 顔やピントがずれた状態,顔の一部が遮蔽した状態など,認識が困難な条件のものを複数含む画像 10 枚を検証した.

評価指標として,正確性と検出率を用いた.正確性は,認識された顔のうち認識結果が正しかった割合(正しく認識された人数/認識された人数),検出率は,画像内に実際に存在する人数のうち正しく認識された割合(正しく認識された人数/実際に存在する人数)を表す.これにより,誤認識と認識漏れの影響をバランスよく評価できる.

評価するしきい値を 0.2,0.3,0.4,0.7 に設定し,各しきい値における顔認識の精度を比較した結果を表 3 に示す.しきい値を高く設定すると,正確性は高く誤認識のリスクは抑えられるが,認識すべき顔の検出率は低く,検出漏れが起こっていることが分かる.例えば,登場人物 4 人が写っている場合でも,1 人しか認識しない場合もあった.一方でしきい値を低く設定した場合,検出率はかなり高いが,誤認識が起こったり未登録の人(通行人やクラスメイトなど)を間違って認識してしまったりすることがある.

この結果を踏まえ,本システムでは検出率と正確性のバランスを考慮し,しきい値を 0.3 に設定した. ドラマでは,クラスメイトや通行人など,画面には映るが主要ではない人物が多くいると想定できるため正確性を優先しているが,しきい値はコード内で自由に設定できるため,登場人物の人数や映像の規模に応じて変更することを推奨する.

しきい値	0.2	0.3	0.4	0.7
正確性	0.8	1	1	1
検出率	1	0.77	0.74	0.17

表 3　正確性と検出率

5.2　GPT-4 の処理時間

「Analyze」を押してから結果が表示されるまでの時間を計測した結果を表 4 に示す.本システムは,動画の視聴中に内容理解の補助をすることを想定しているため,処理時間が長いと視聴者のストレスになりかねない.動画の内容やセリフ量にも依存するが,この結果によると数十秒で解析が終了しており,現状では視聴者の大きなストレスにはならないと考えられる.また,GPT-4 による解析の前にもとの台本の不要な部分を切り取っているため,ネタバレを防ぐと同時に処理時間の短縮にも繋がっている.しかし,台本の文字数や情報量が増えるほど処理時間が長くなり,特に長編ドラマやセリフ量が多い作品では,視聴体験に影響を与える可能性がある.

動画の長さ	15m(約 2000Tokens)	30m(約 4100Tokens)
処理にかかる時間	15s	25s

表 4　GPT-4 の処理時間

5.3　生成された文章の精度

GPT-4 が生成した解析結果の文章の内容の精度や一貫性を評価するため，被験者 26 名による約 2 分のショートドラマの解析文章評価を行った.台本との一致度やネタバレの有無,明らかに誤った情報がないか,不自然な文章ではないかということなどを踏まえ,文章の正確性とリアルタイム性をそれぞれ 5 段階で評価した.その結果,正確性が 3.7,リアルタイム性が 4.0 となり,「あらすじが簡潔にまとまっていてわかりやすい」という肯定的な意見が多く見られた.しかし,生成 AI 特有の不自然な文章も一部に見られるという指摘もあった.

さらに,50 分程度のドラマの 15 シーンを自己検証した.こちらも明らかな誤情報やネタバレはほぼ見られなかったため,本システムで生成される文章は,重要な内容を捉え正確に表現しており,視聴者が内容や登場人物の関係性を理解するうえで十分な精度であると判断した.ただし,複雑な内容が含まれる場合,若干の曖昧さが生じる可能性もある.

図 7 桃太郎の解析結果 戦いシーン前(上) 戦いシーン後(下)

第 6 章　まとめと今後の展望

本研究では,顔識別モデル InsightFace や生成 AI を用いて,動画の視聴体験を向上させる Web システムを提案した. 解析時点以降の内容を除外するという仕組みを導入することで,視聴中にネタバレを防ぎながら情報を動的に提供するシステムを実現した.これにより,視聴動画の登場人物や内容の把握が容易になり,物語への理解度や没入感を高めることが期待できる結果となった.特に,長編ドラマの新規視聴者や複雑なストーリーの理解が困難な視聴者にとっては,最新のネタバレを受けることなく情報を受け取ることができ,非常に効果的であるだろう.

今後の展望としては,4.2 で取り上げた GPT-4 による解析速度の更なる短縮に取り組みたい. そのため,必要に応じ解析範囲を限定する機能や,重要なシーンを自動的に優先して解析するアルゴリズムなどの導入も有効であると考える.システムの快適な利用を実現するためには,解析速度と内容の精度をバランスよく保つことが重要である.

また,現在は GPT-4 が出力した回答をそのまま表示しているため,時々著しく精度の低い回答が提供される場合がある.このような課題を解決するため,GPT-4 が生成した回答が十分な精度であるかを GPT-4 が再評価したうえで,最終結果として出力するような機構の導入も有効ではないかと考えている.

このシステムの利用を通し,ドラマなどの映像作品に触れる体験がより豊かで快適なものとなることを目指し,引き続き改良を重ねていきたい.

謝辞

本研究及び論文の作成にあたり，多くのご指導，ご助言を頂きました静岡大学工学部の岡部誠准教授に心から感謝申し上げます．また，ご助力頂いた修士課程学生及び学部生の皆様に深く感謝致します．

参考文献

[1] Midjourney(オンライン), 入手先< https://www.midjourney.com/home >（参照 2025/02/10)

[2] Pika(オンライン), 入手先< https://pika.art/ >

[3] Yunxin Li, Haoyuan Shi, Baotian Hu, Longyue Wang, Jiashun Zhu, Jinyi Xu, Zhen Zhao, Min Zhang, “Anim-Director: A Large Multimodal Model Powered Agent for Controllable Animation Video Generation”, In SIGGRAPH Asia 2024

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, Ross Girshick, “Segment Anything”, In ICCV 2023

[5] Florian Schroff, Dmitry Kalenichenko, James Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering”, In CVPR 2015

[6] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf, “DeepFace: Closing the Gap to Human-Level Performance in Face Verification”, In CVPR 2014

[7] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, Wei Liu, “CosFace: Large Margin Cosine Loss for Deep Face Recognition”, In CVPR 2018

[8] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, Le Song, “SphereFace: Deep Hypersphere Embedding for Face Recognition”, In CVPR 2017

[9] InsightFace（オンライン), 入手先〈 https://insightface.ai/ 〉(参照 2025-02-06).

[10] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, Stefanos Zafeiriou, “RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild”, In CVPR 2022.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Microsoft Research, “Deep Residual Learning for Image Recognition”, In CVPR 2016.

[12] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou, “ArcFace: Additive Angular Margin Loss for Deep Face Recognition”, In CVPR 2019.

[13] OpenAI, “GPT-4 Technical Report”, 2023（オンライン), 入手先〈 https://arxiv.org/abs/2303.08774 〉(参照 2025-02-06).

[14] ClovaNote（オンライン), 入手先〈 https://clovanote.line.me/ 〉(参照 2025-02-06).

[15] FFmpeg:Documentation（オンライン), 入手先〈 https://ffmpeg.org/documentation.html 〉(参照 2025-02-06).

[16] Hao Zeng, Zhiyong Zhang, Lulin Shi, “Research and Implementation of Video Codec Based on FFmpeg” In ICNISC 2016