令和七年度　卒業論文

手書き文字認識モデルエンコーダへの言語特徴付与による有効性

大阪工業大学

ロボティクス＆デザイン工学部

システムデザイン工学科

学籍番号　922022

氏　名　　工藤 滉青

指導教員　瀬尾 昌孝

目次

図目次

表目次

1. 序論

1.1. 研究背景と目的

本研究では、手書き文字認識（Handwritten Text Recognition: HTR）を対象とする。HTR は、手書き文字画像から対応する文字列を自動的に認識する技術であり、手書き文字の帳票処理、文書アーカイブ、業務文書の電子化など幅広い分野で利用されている。近年の HTR では、入力画像をエンコーダによって特徴ベクトルへ変換し、その特徴ベクトルをもとにデコーダが文字列を予測する end-to-end の深層学習モデルが広く用いられている。学習時には、予測結果と正解ラベルとの間で損失を計算し、モデル内部のパラメータを最適化する。これによって、モデルの予測がより正しい文字列出力へ修正されていく。

しかし、既存の HTR モデルでは視覚特徴取得のみに基づいたエンコーダが中心であり、字形の曖昧さや画像の欠損・損傷に対して誤認識が生じやすい。そこで、Connectionist Temporal Classification(CTC)に基づく認識結果に対して外部の言語モデルを用いたデコードを行うことによって、文全体の構文的一貫性や語彙的整合性といった言語的側面を補完する手法が広く用いられている。一方で、言語知識は主に推論時の補正として利用され、学習段階でエンコーダの特徴抽出能力に十分反映されていないという課題がある。

本研究では、LLM による中間損失を導入し、文全体の構文的一貫性や語彙的整合性といった言語的側面を、視覚特徴と同時にエンコーダで学習可能な枠組みを提案する。

1.2. 論文構成

第2章では、本研究でエンコーダモデルとして使用する TrOCR、自動音声認識における LLM による中間損失を用いた言語特徴付与について述べる。第3章では、提案手法の概要、提案手法に使用する Connector Layer、Language-Aware Intermediate Loss そして CTC decoder について述べる。第4章では、実験の設定、概要について述べるとともに、これに基づいて得られた結果をまとめ、それについて考察した内容を第5章でまとめた。最後に、この研究全体を第6章にまとめた。

2. 関連研究

本章では、本研究の関連研究について述べる。本研究では、HTR を行うにあたり、ベースモデルのエンコーダとして TrOCR を採用した。また、自動音声認識分野においてエンコーダ層の中間層出力に対して、LLM を導入することで認識精度の向上を実現した枠組みを参考にした。したがって、本章ではまずベースモデルのエンコーダとして採用した TrOCR の構造について説明する。次に、自動音声認識における LLM の活用手法について述べる。

2.1. TrOCR による手書き文字認識手法

TrOCR は、Transformer-based OCR モデルの略称である[1]。従来の手法では、CNN による特徴抽出と RNN による系列変換を組み合わせた構造が主流であった[2,3]。その後、Transformer[4]アーキテクチャを活用することで大幅な改善が見られた。そこで、TrOCR は、エンコーダに事前学習済みの ViT 系モデル、デコーダに BERT 系モデルを採用した Encoder-Decoder 構造を持ち、end-to-end で文字認識を行う。図 2.1 に TrOCR のモデル構造を示す。

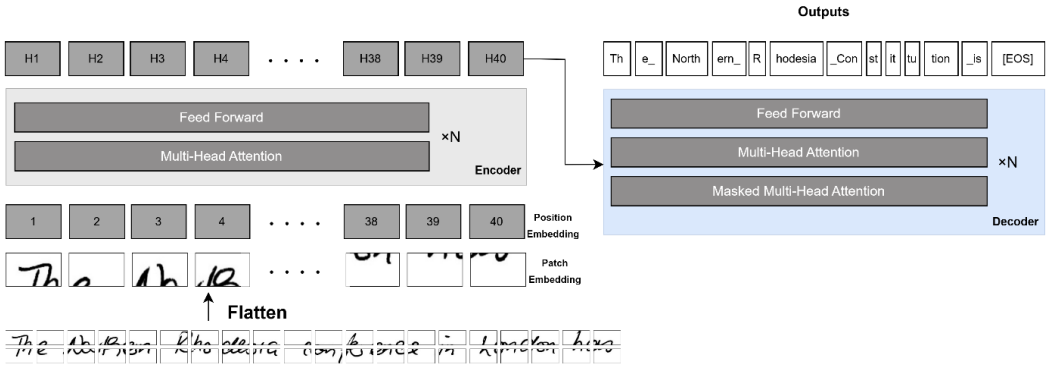

図 2.1 TrOCR モデル

エンコーダでは、入力画像を N パッチに分割し、各パッチを線形射影することで埋め込みベクトルを生成する。これに位置埋め込みを加え、Transformer ブロックに入力することで画像特徴を抽出する。デコーダでは、エンコーダから得られた特徴ベクトルをもとに、言語モデルデコーダによってテキストを生成する。TrOCR の大きな特徴は、事前学習済みモデルを活用できる点にある。エンコーダには画像分類タスクで事前学習された ViT 系モデルを、デコーダには BERT 系の事前学習済みテキスト Transformer を初期値として用いて、さらに大規模合成手書き文字画像データを用いた再学習を行う。これによって得たモデルを対象タスクの少量データ(数千枚規模)でパラメータをファインチューニングすることによって、対象データの文字種や筆跡に適応し、学習データが数千枚画像のような少量のデータでも高い認識精度を達成できる。

なお TrOCR のデコーダに言語モデルを採用した理由として、論文では従来の HTR モデルが CTC と外部言語モデルを組み合わせた後処理によって精度向上を図ることが多い点を指摘している。その上で TrOCR では、デコーダ内部に BERT 系のモデルの事前学習で獲得された文脈に基づく文字列の妥当性判断といった言語モデル能力が組み込まれることによって外部言語モデルを不要にできることを挙げている。

本研究では、TrOCR エンコーダを基盤としたモデルをベースラインモデルへ取り入れた。

2.2. 自動音声認識における LLM の活用

自動音声認識では、従来は音声特徴の抽出・アライメント学習に重点が置かれ、言語的な依存関係の活用は限定的だった。特に、CTC に基づく自動音声認識は非自己回帰で高速に推論できる。しかしながら、CTC デコーダでは長距離の言語依存を特徴としてとらえづらいという課題がある。そこで近年は、LLM が持つ大規模な言語知識を自動音声認識に注入し、CTC の一般的に言語的制約を明示的に取り込みにくいという欠点の補完に関する研究が進んでいる。本節では Altinok (2025) が提案した Language-Aware Intermediate Loss (LAIL) に基づく枠組みを取り上げる[5]。CTC ベースの自動音声認識における中間層の特徴ベクトルを、Connector Layer を用いて LLM の埋め込み空間へ写像する点が特徴である。学習時には、この写像結果に対して LLM の中間損失 Causal Language Modeling Loss (CLM loss) を中間損失として与える。これにより、CTC の計算効率を維持しながら、認識性能の向上を図る枠組みを提案している。図 2.2 にこの手法のモデル構造を示す。

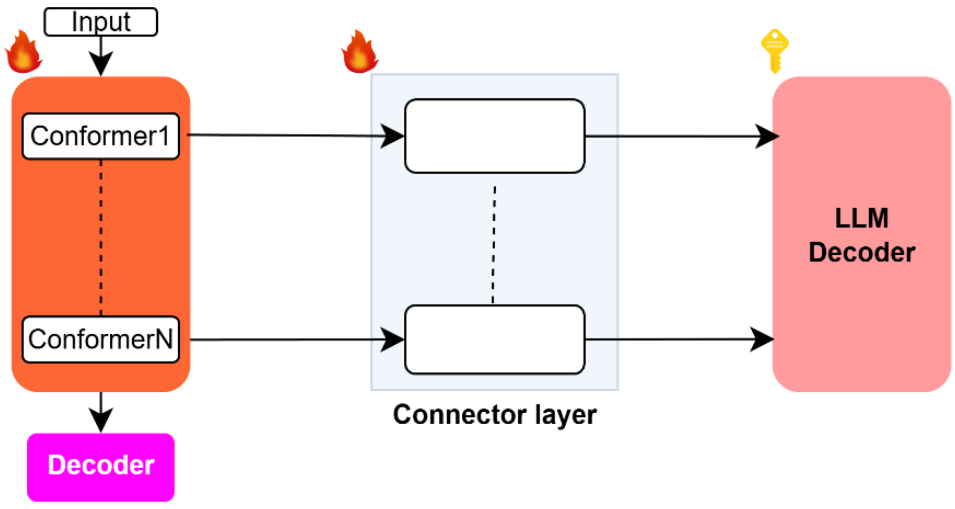

図 2.2 自動音声認識モデル

この手法では、音声認識モデルの特定のエンコーダ中間層に Connector Layer を接続し、これによって音声特徴から言語特徴への変換、LLM の入力次元との整合を行っている。そして、正解

テキストと LLM の出力（次トークンの条件付き確率）との間で CLM loss を計算する。CLM loss とは、式(2.1)に示すように、第lの Connector Layer の出力$\boldsymbol{h}$と、系列中の各位置tにおいて、それ以前のトークン列$\boldsymbol{x_{<t}}$のみを条件として次トークン$\boldsymbol{x_t}$を予測する確率を最大化する（すなわち、その負の対数尤度を最小化する）ための損失である。

$$L_{CLM} = -\sum_{t=1}^{T} \log P_{LLM}\left(\boldsymbol{x_t}|\boldsymbol{x_{<t}}, \boldsymbol{h^l}\right) \tag{2.1}$$

さらに、この CLM loss を用いて、選択した中間層集合Sに含まれる各層lに対して CLM loss を計算する。そして、それらの CLM loss をハイパーパラメータ$\boldsymbol{\lambda}$により調整し、式(2.2)に示す Language-Aware Intermediate Loss(LAIL)を計算することで、学習時に LLM が有する言語知識を音声認識モデルエンコーダへと取り込む。

$$L_{LAIL} = -\sum_{l \in S} \lambda_l L_{CLM}^{(l)} \tag{2.2}$$

この手法の特徴は、学習時にのみ LLM を使用し、推論時には従来の CTC デコーダをそのまま用いる点にある。これにより、CTC の高速な推論という利点を維持したまま、LAIL を導入して言語的知識を学習に取り込むことで、音声認識における評価指標である WER（Word Error Rate）を改善した。特定コーパスでは WER が 5.1 から 3.6 に低下し、相対的に最大約 29%の改善が報告されている[5]。

本研究では、この枠組みを参考に、手書き文字認識において LLM の言語知識を活用する手法を提案する。

3. 提案手法

本章では、本研究の提案手法について述べる。3.1 節では提案手法の全体像について述べる。3.2 節では中間層の出力を LLM へ渡すための Connector Layer、3.3 節では Language-Aware Intermediate Loss、3.4 節では CTC decoder の詳細について述べる。

3.1. モデル構造

本研究が対象とする HTR では、入力として手書き文字列を含む画像を与え、出力として対応する転写テキスト（文字列）を推定する。

本研究では、TrOCR のエンコーダをベースモデルのエンコーダとして採用する。TrOCR は大規模データによる事前学習モデルとして公開されており、ファインチューニングが容易であるため広く採用されている。また、DTrOCR をはじめとする TrOCR 系の派生研究[6,7]や、PARSeq[8]などの Transformer-based HTR モデルにおいて、TrOCR は基盤モデルあるいは比較対象として広く採用されている。そのため、本研究で TrOCR を用いることで、提案手法を既存研究と公平に比較できるだけでなく、他の HTR モデルへの拡張可能性や汎用性についても議論しやすいという利点がある。

ただし、本研究では、LLM に基づく中間損失の効果を明確に検証するため、TrOCR のデコーダのように言語モデルに基づいて文字列を生成する方式ではなく、CTC デコーダを適用した TrOCR-CTC を採用する。TrOCR のデコーダは言語モデルを使用しており、それ自体が強い言語的帰納バイアスを持つため、LLM に基づく中間損失による言語認識特徴の導入が性能に与える寄与を分離して評価しにくい。一方、CTC デコーダ単体では言語的制約が相対的に弱く、HTR モデルのエンコーダで特徴抽出する際の影響をより直接的に観測できる。

以上より、本研究では、TrOCR をベースモデルのエンコーダとして用いることによって HTR における LLM に基づく中間損失の効果を明確に検証する。また、文字列予測を行う TrOCR デコーダは用いずに CTC デコーダへ置き換えた TrOCR-CTC を採用する。

提案モデルでは、TrOCR エンコーダの中間層出力に Connector Layer を接続し、中間特徴を LLM の埋め込み空間へ写像する。写像後の特徴ベクトルを用いて Language-Aware Intermediate Loss を計算し、CTC 損失（3.4 節）と併せて学習することで、視覚的に曖昧な字形に対しても文脈的に整合する特徴を抽出するよう促す。なお、LLM の重みは固定とし、LLM は学習時の損失計算のみに用いる。提案手法の全体構造を図 3.1 に示す。

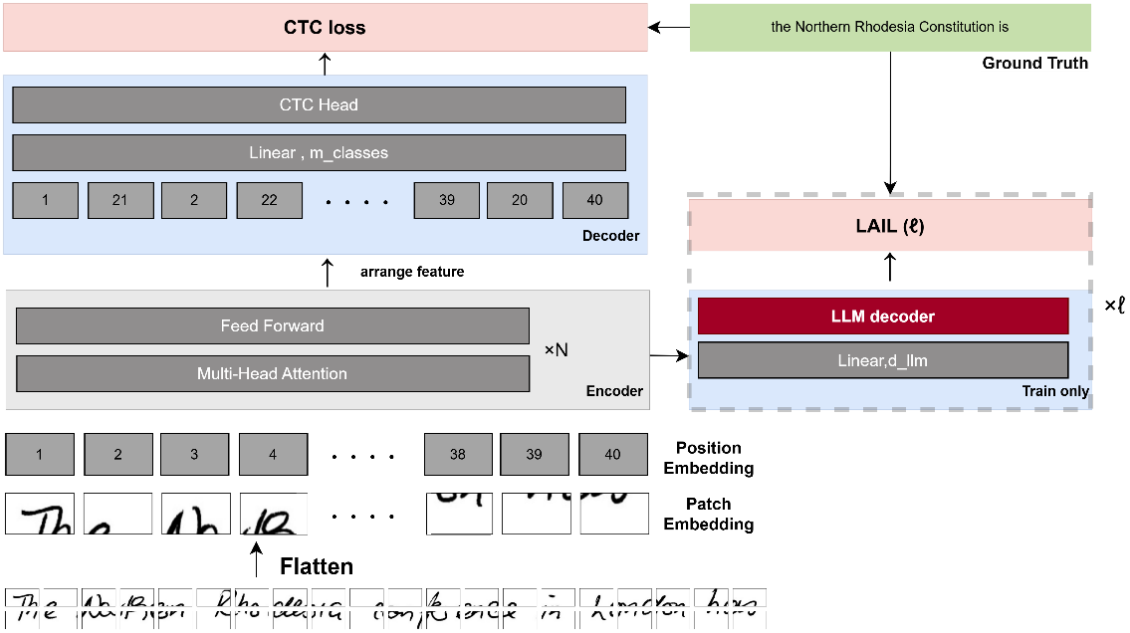

図 3.1 LLM を用いた TrOCR の提案モデル

3.2. Connector Layer

提案手法に用いる Connector Layer の役割は2つある。1つ目は、画像エンコーダの中間層出力の次元数とLLMの埋め込み空間の次元数を合わせること。2つ目は、画像の特徴空間から言語の特徴空間へと写像すること。

TrOCR のエンコーダは 12 層の ViT ブロックから構成されている。提案手法では、特定の中間層の出力を取り出し、Connector Layer へ入力する。エンコーダ中間層の出力は隠れ状態の次元d_{enc}を持つ。一方、LLM の埋め込み空間は次元d_{llm}を持つ。Connector Layer は式(3.1)で構成される。

$$\boldsymbol{h}_{\text{proj}} = \boldsymbol{W} \cdot \boldsymbol{h}_{enc} + \boldsymbol{b} \tag{3.1}$$

ここで、$\boldsymbol{h}_{enc} \in \mathbb{R}^{\boldsymbol{d}_{enc}}$はエンコーダ中間層の出力、$\boldsymbol{W} \in \mathbb{R}^{\boldsymbol{d}_{llm} \times \boldsymbol{d}_{enc}}$は重み行列、$\boldsymbol{b} \in \mathbb{R}^{\boldsymbol{d}_{llm}}$はバイアス、$\boldsymbol{h}_{proj} \in \mathbb{R}^{\boldsymbol{d}_{llm}}$は写像後の出力である。

学習時には、Connector Layer と TrOCR のエンコーダを同時に学習する。これにより、LLM の持つ言語知識を保持したまま、エンコーダが言語的に意味のある特徴を学習できるようにする。

3.3. Language-Aware Intermediate Loss

LAIL は、LLM が有する言語知識を学習過程においてエンコーダの中間特徴へ反映させるための損失関数である。本損失を導入することで、エンコーダは視覚的特徴だけでなく、文全体の構

文的一貫性や語彙的整合性といった学習を促す。3.2 節で述べた Connector Layer により、エンコーダ中間層の出力は LLM の埋め込み空間へ写像される。この写像された特徴ベクトルから正解テキストとの間で CLM loss を計算する。CLM loss は、系列中の各位置tにおいて、それ以前のトークン列のみに基づいて次トークンを予測する言語モデルの損失であり、式(3.2)のように定義される。

$$\mathcal{L}_{CLM} = -\sum_{t=1}^{T} \log P(\boldsymbol{y}_t \mid \boldsymbol{y}_{<t}, \boldsymbol{h}_{proj}) \tag{3.2}$$

ここで、Tは正解テキストのトークン数、$\boldsymbol{y}_t$は時刻tにおける正解トークン、$\boldsymbol{y}_{<t}$は時刻tより前のトークン列、$\boldsymbol{h}_{proj}$は Connector Layer の出力である。

さらに、この CLM loss を用いて、選択した中間層集合Sに含まれる各層lに対して計算される CLM loss をハイパーパラメータλにより調整した。式(3.3)に示す Language-Aware Intermediate Loss を計算することで、学習時に LLM が有する言語知識を HTR モデルへと取り込む。

$$\mathcal{L}_{LAIL} = -\sum_{l \in S} \lambda_l L_{CLM}^{(l)} \tag{3.3}$$

最終的な学習時の損失関数は、TrOCR の 3.4 節で述べる CTC 損失$\mathcal{L}_{CTC}$と LAIL を組み合わせた式(3.4)のようになる。

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{CTC} + \alpha \mathcal{L}_{LAIL} \tag{3.4}$$

この損失を最小化することで、エンコーダが LLM の言語知識も反映するように学習される。また、$\mathcal{L}_{LAIL}$を調整するための α はハイパーパラメータである。

3.4. CTC decoder

提案手法では、TrOCR エンコーダの出力によって得られた特徴ベクトルに対して縦方向の mean pooling を適用し、横方向の系列特徴へ下の図 3.2 のように集約する。

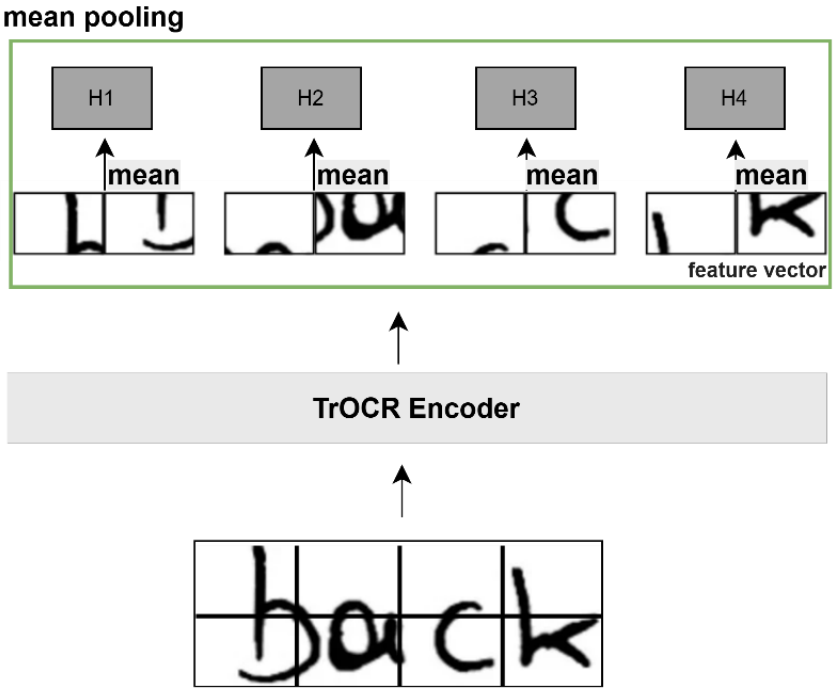

図 3.2 mean pooling

その後、CTC に基づくデコーダで文字列を推定する。エンコーダ出力である$\mathbf{E} \in \mathbb{R}^{B\times H\times T\times D}$（$B$：バッチサイズ、$H$：縦方向、$T$：横方向の時系列長、$D$：特徴ベクトル次元）を、縦方向 mean pooling により各時刻の特徴ベクトル$\mathbf{z}_{b,t} \in \mathbb{R}^{D}$を

$$\mathbf{z}_{b,t} = \frac{1}{H}\sum_{h=1}^{H} \mathbf{E}_{b,h,t,:} \tag{3.5}$$

として得る。縦方向に mean pooling を行う理由は、文字が存在しない余白領域の特徴が、そのまま後段の Linear 層に入力されると、背景部分からも文字クラスへの投票が発生してしまうためである。このような背景由来の投票が混ざると、同一の文字に対応する時刻付近で予測が不安定になり、CTC の復号時に不要な空白トークンや誤った文字が挿入されるなどの誤認識が起きやすくなる。そこで縦方向に平均化して特徴を統合することで、背景領域の影響を相対的に弱め、横方向の系列特徴をより安定にし、CTC における不要な挿入を抑制する。

mean pooling 後の特徴ベクトルに対し、Linear 層により各時刻ロジットを計算する。クラス数をC（文字集合＋〈*blank*〉を含む）、$\mathbf{W} \in \mathbb{R}^{C\times D}$とすると、

$$\mathbf{h}_{b,t} = \mathbf{W}\mathbf{z}_{b,t} + \mathbf{b} \tag{3.6}$$

である。ここで$\mathbf{h}_{b,t} \in \mathbb{R}^{C}$は、時刻$t$における各クラス$C$に対応するロジットを表す。このロジットに対してクラス次元に softmax を適用することで、各クラスの対数確率を次式で与える。

$$\log \mathbf{p}_{b,t} = \log \mathrm{softmax}\left(\mathbf{h}_{\mathbf{b,t}}\right) \tag{3.7}$$

CTC では、時刻tごとに 1 文字を出力する長さTのラベル列をアライメント列$\pi = (\pi_1, \dots, \pi_T)$と呼ぶ。ここで$\pi_t$は時刻$t$に選択された集合$C$内のクラスを表す。入力$\mathbf{x}$に対するアラインメント$\pi$の確率と目的文字列$\mathbf{y}$の確率は、時刻ごとの独立仮定のもと次式で与えられる。

$$P(\boldsymbol{\pi} \mid \mathbf{x}) = \prod_{t=1}^{T} p_t(\pi_t \mid \mathbf{x}) \tag{3.8}$$

$$P(\mathbf{y} \mid \mathbf{x}) = \sum_{\boldsymbol{\pi} \in B^{-1}(\mathbf{y})} P(\boldsymbol{\pi} \mid \mathbf{x}) \tag{3.9}$$

ここで、写像$B(\cdot)$はアライメント列πに対して連続重複の縮約と<blank>の除去を行い$B^{-1}(\mathrm{y})$は$B(\pi)$を満たすすべてのアライメント列の集合を表す。学習時には、負の対数尤度として次式の CTC 損失を最小化する。

$$\mathcal{L}_{\mathrm{CTC}} = -\log P(\boldsymbol{y} \mid \boldsymbol{x}) \tag{3.10}$$

推論時は、各時刻で最大確率のクラスを選択した系列に対して写像$B(\cdot)$を適用し、連続重複の縮約と<blank>除去により最終的な認識結果を得る。

4.実験

本章では、提案手法の有効性を検証するために実施した実験について、実験設定および実験の概要を述べる。まず 4.1 節では、学習、評価に用いたデータセットの詳細について示す。続く 4.2 節では、性能評価に用いる指標を説明する。4.3 節では LLM を用いないベースラインの実験設定を整理し、4.4 節では提案手法の実験設定を示す。最後に 4.5 節では、本提案手法の有効性を示すために行う実験の概要を述べる。

4.1. データセット

本研究では、手書き英単語・文章から構成される IAM Handwriting Database[9]を用いた。IAM Handwriting Database は、英語の手書き文字を収録した代表的なオフライン HTR 用データセットであり、複数筆者による手書き文字画像とその転写テキスト（ground truth）から構成される。IAM には Form（文書）・Line（行）・Word（単語）といった複数粒度のアノテーションが提供されており、文書画像から行・単語へ分割した画像と対応する転写が付与されている。データセット全体としては、文字クラス数は 79 文字、1,539 枚の手書きフォームが 657 名の筆者によって記述されている。

本研究ではこのうち Line レベル画像を入力として用い、さらに画像のアスペクト比を保ちながら画像サイズを高さ 384px、幅 2048px にリサイズした 1 行画像に対応する転写テキストを教師信号として学習・評価を行った。

データの分割には、Aachen split[10]を採用した。Aachen split では、同一筆者がテストデータと検証データ両方に入らないように分割されるため、筆者依存の過学習を避けた汎化性能を評価できる。本研究では IAM の行データを Aachen split に従って学習データ、検証データ、テストデータに分割し、以降の実験はこの分割に基づいて実施した。分割後の各集合の行数は

- 学習データ :6161 枚
- 検証データ :966 枚
- テストデータ :2915 枚

である。

4.2. 評価指標

評価指標には、CER(Character Error Rate)を用いた。CER は予測文字列p_iと正解文字列g_iの Levenshtein 距離$d(p_i, g_i)$を用いて、次式により算出する。

$$\mathrm{CER} = \frac{\sum_{i=1}^{N} d(p_i, g_i)}{\sum_{i=1}^{N} \max(1, |g_i|)} \tag{4.1}$$

N は評価サンプル数、$d(p_i, g_i)$が Levenshtein 距離を測る挿入・削除置換の最小回数で$|\ g_i\ |$正解文字列の文字数、$\max(1, |g_i|)$は、正解が空文字のときに 0 除算を避けるための処理である。

4.3. ベースラインの実験設定

本研究では、3 章で述べた提案手法モデルに LAIL を導入しないものをベースラインモデルとする。ベースラインモデルの学習時の設定について述べる。最適化手法には AdamW を用いた。本研究で用いた主なハイパーパラメータを表 4.1 に示す。

表 4.1 ベースラインの実験設定

バッチサイズ	4
エポック数	200
学習率	1×10^{-4}
乱数シード	42

損失値の推移および検証データにおける精度（CER）の変化を参照し、性能改善が頭打ちとなった時点で学習を終了した。

4.4. 提案手法の実験設定

提案手法モデルの学習時の設定について述べる。最適化手法には AdamW を用いた。本研究で用いた主なハイパーパラメータを表 4.1 に示す。

表 4.2 提案手法モデルの実験設定

バッチサイズ	4
学習率	1×10^{-4}
乱数シード	42
$\mathcal{L}_{LAIL}$の重み	0.01

損失値の推移および検証データにおける精度（CER）の変化を参照し、性能改善が頭打ちとなった時点で学習を終了した。なお、本研究では学習の打ち切りは概ね 200 エポック前後で収束傾向が見られた時点を目安とした。

4.5. 実験内容

本節では、実験内容について述べる。はじめに、LLM に基づく中間損失による言語認識特徴の導入が性能に与える寄与を明確にするために、3 章で述べた提案手法モデルと LAIL を用いないベースラインモデルを比較した。ベースラインモデル、提案手法モデルともに認識精度を 4.2 節で述べた CER で定量的に評価した。

次に、LLM の埋め込み空間へ入力するエンコーダ出力について、Connector Layer の数と配置によって、LLM が有する言語知識を学習過程において効果的にエンコーダの中間特徴へ反映させることができるのかを検証した。具体的には以下のような検証をおこなった。

- **3 層**:4,8,12 の層目の出力、10,11,12 層目の出力を使用
- **2 層**: 異なるエンコーダ層を対象とした組み合わせを比較
  - 最下層+最上層: 3、12 層目
  - 中間層+最上層: 6、12 層目
  - 上位層+最上層: 9、12 層目
- **1 層**:12 層目の出力のみを使用

最後に、使用する LLM の性能によって言語特徴の付与の効果が変化するのかを検証する。そのため、LLM の性能ごとの精度を比較した。

なお、学習設定は 4.3、4.4 節で述べた設定および評価指標は 4.2 節で述べた指標を用い全条件で統一し、LLM の有無・接続層構成・LLM サイズのみを変更した。

また、すべての実験は単一の GPU（NVIDIA GeForce RTX 4060 Ti, 16GB VRAM）で実施した。

5. 結果と考察

本章では、第4章の実験概要で述べた実験結果を示し、提案手法の有効性と各要因の影響について考察する。まず 5.1 節では、LLM を用いないベースラインモデルと提案手法モデルの性能を比較し、LAIL の導入が認識精度に与える効果を明らかにする。次に 5.2 節では、Connector Layer の数と配置を変化させた結果を整理し、どの層を入力とすることが有効であるかを考察する。最後に 5.3 節では、使用する LLM のモデル性能（1B/3B）の違いによる性能差を比較し、LLM の性能による認識性能への影響について議論する。なお、以降の実験の設定、評価指標は特に断りがない場合第4章で述べた設定を採用する。

5.1. ベースラインモデルとの比較

表 5.1は、IAM データセットにおける CER の結果を示しており、3章で述べたベースラインモデルと Connector Layer を 11,12 層目に接続して学習させた提案手法モデルとを比較した結果である。また、使用した LLM は 1B パラメータを持つ Llama 3[11]モデルを使用して取得された。

表 5.1 ベースラインモデルと提案手法の結果

ベースラインモデル(CER:%)	提案手法モデル(CER:%)
13.64	10.77

表 5.1より、ベースラインモデルに対して提案手法のほうが CER が減少したことがわかる。

次に、具体的な出力の変化を確認するために 5 つのテストデータ内のサンプル画像を用いた、ベースラインモデルと提案手法モデルそれぞれの出力例を図 5.1～図 5.5 へ示す。また、図の赤字はモデルのテキスト出力における誤認識箇所である。

正解テキスト　　:the switch because of the topicality of African

ベースラインモデル　　:the suith beause ot the toicaliy of Africann

提案手法モデル　　:the switch because of the topicality of African

図 5. 1″the switch because of the topicality of African″

正解テキスト　　:to make the strongest criticisms. He said

ベースラインモデル　　:to make the strongest caiticisns. He said

提案手法モデル　　:to make the strongest criticisms. He said

図 5. 2“to make the strongest criticisms. He said″

正解テキスト　　:maturity. One remembered that he was a lawyer

ベースラインモデル　　:mnaturrty. Ore remembered tthat he wos a lowyer

提案手法モデル　　:maturity. One remembered that he was a lawyer

図 5. 3“maturity. One remembered that he was a lawyer”

正解テキスト　　:come what may, we are never alone when

ベースラインモデル　　:comse wlat may, we are never obome wlren

提案手法モデル　　:come what may, we are never alone when

図 5. 4“come what may, we are never alone when”

正解テキスト　　:Sentence Database P03-189

ベースラインモデル　　:intennce Datahase　P03-y89

提案手法モデル　　:intenceDatahase　Po3. Yg9

図 5. 5″Sentence Database P03-189″

図 5.1〜図 5.5 より、ベースラインモデルの予測では、入力画像中の字形の曖昧さや画像の欠損・損傷による誤認識が生じていることが確認できる。例えば図 5.4 では、“what”に含まれる筆記体の“h”が筆記体の“l”と形状的に酷似しており、1 章で述べたように、視覚特徴取得のみに基づいたエンコーダでは判別が難しいため、誤認識が発生したと考えられる。

一方、提案手法モデルでは“what”の“h”を正しく認識できており、前後の単語である“come”と“may”についても正しく出力できている。これは、LAIL の導入により LLM が有する言語知識を学習過程においてエンコーダの中間特徴へ反映させるためエンコーダに言語的特徴が付与されることで、LLM で事前学習された言語知識（例：“come what may”のような定型表現や語の典型的な言い回し）を手がかりとして、入力画像中の字形の曖昧さや画像の欠損・損傷を文脈から補正できるような特徴をエンコーダで獲得できたためだと考えられる。

ただし、図 5.5 のように“P03-189”といった記号列・固有表現に対しては、提案手法モデルの出力が“Po3. Yg9”のようにベースラインモデルより誤りが増える例も一部確認された。これは、LLM の学習データ内で同様の表記が十分に学習されていない場合、上記の提案手法の強みがかえって不利に働き誤った文字の予測となったと考えられる。

5.2. Connector Layer の数と配置の効果

次に、Connector Layer の数と配置が及ぼす影響を、様々な構成で実験することで調査した。1〜3 層配置を比較した下の表 5.2 にその結果をまとめた。また、使用した LLM は 1B パラメータを持つ Llama 3 モデルを使用して取得された。なお、4.5 節で述べた使用したハードウェアのメモリの制約上 Connector Layer を 3 層つけた実験設定はバッチサイズを 2 で実験をした。

表 5.2 Connector Layer の配置が CER に与える影響

Connector Layer	CER(%)
10,11,12	12.54
4,8,12	11.96
3,12	12.55
6,12	13.16
11,12	10.77
12	12.44

- **3 層**：3 層配置では、4,8,12 層（CER 11.96%）が最も良好であり、これに対して 10,11,12 層（12.54%）は 0.58 ポイント性能が低下した。上位層に Connector Layer を 3 層密集させる構成では、LLM 由来の言語特徴の注入が強まりすぎ、視覚特徴との干渉が増えることで性能が低

下する可能性がある。また、データ量が限られる場合には、Connector Layer 数増加に伴う学習可能パラメータの増加により過学習が生じやすくなる可能性も考えられる。一方、等間隔に配置した4,8,12層が比較的良好であったことは、視覚特徴の保持と段階的な言語特徴の統合が両立しやすい可能性を示唆する。なお、3 層構成のみバッチサイズを 2 としているため、他条件(バッチサイズ 4)との厳密な比較には留意が必要である。

- **2 層**:2 層配置では、11,12 層(10.77%)が最も良好であり、3,12 層(12.55%)や 6,12 層(13.16%)よりも高い性能を示した。これは、上位層ほど表現がより文脈的・統合的となり、LLM 由来の言語特徴を有効に取り込みやすい段階にある可能性を示唆する。
- **1 層**:1 層配置 12.44%)でもベースラインと比較して改善が得られたことから、言語特徴を HTR モデルへ付与すること自体は有効である。

以上より、Connector Layer は単に数を増やせばよいのではなく、本実験条件では 2 層(特に 11,12 層)配置が最も効果的であることが示された。これは、上位層ほど表現がより文脈的・統合的となり、LLM 由来の言語特徴を有効に取り込みやすい段階にあるためだと考えられる。一方で、層数を増やして上位層に密集させる配置では、LLM 由来の言語特徴の注入が強まりすぎ、視覚特徴との干渉が増えることで性能が低下する可能性があり、さらにデータ量が限られる場合には過学習の影響も受けやすい点に留意が必要である。

5.3. LLM の性能による効果

LLM の性能による影響を評価するために、1B、3B パラメータの LlaMa モデルを用いた場合の精度を比較した。これらの実験では、Connector Layer 数は 12 層に固定し結果は表 5.3 にまとめられている。

表 5.3 LLM 性能差による効果

LLM	CER(%)
1B	12.44
3B	12.66

表 5.3 より、本実験設定では 3B モデルは 1B モデルに比べて CER がわずかに増加し、性能改善は確認されなかった(1B: 12.44%、3B: 12.66%)。差は 0.22 ポイントと小さいものの、LLM の規模拡大が CER 改善に直結しない可能性が示唆される。要因の一つとして、学習データ規模に対して 3B モデルの容量が大きく、過学習が生じやすかった可能性が考えられる。したがって、より大規模な LLM を用いる場合は、データ量の拡充を検討する必要がある。

6. 結論

本研究では、視覚特徴のみに基づいて学習されるエンコーダを用いる既存手法において、言語的特徴を十分に取り込めないことが原因で HTR の認識精度が低下するという課題に着目した。LLM による中間損失を導入し、文全体の構文的一貫性や語彙的整合性といった言語的側面を、視覚特徴と同時にエンコーダで学習可能な枠組みを提案した。具体的には、エンコーダの所定の層に Connector Layer を接続し、その出力を Llama 3 の埋め込み入力として与える。さらに、LLM の出力に基づいて LAIL を算出し学習に用いることで、HTR の認識性能向上を図った。

LLM による中間損失を導入し、文全体の構文的一貫性や語彙的整合性といった言語的側面を、視覚特徴と同時にエンコーダで学習可能な枠組みを提案した。具体的には、選択したエンコーダ層に Connector Layer を接続し、その出力を Llama 3 の埋め込み入力としてその出力の値から LAIL を計算し使用することで HTR の認識性能向上をはかった。

IAM データセットを用いた実験では、Connector Layer を適切な層に配置した設定において CER がベースラインから改善し（13.64%→10.77%）、相対で約 21%低減した。これにより、提案手法が HTR の認識性能向上に有効であることを確認した。一方で、LLM の規模を大きくすることや Connector Layer 数を増やすことが CER の単調な低下に直結するわけではなく、一定の条件下では性能が頭打ち、あるいは悪化する場合も見られた。要因として、データ量に対してモデル容量が過大となることで過学習が生じる可能性や、LLM 由来の言語特徴の注入が強まりすぎた結果、視覚特徴との干渉が生じたと考えられる。したがって、モデル構成だけでなく、データ量の確保や正則化、学習条件の調整といった学習設定にも留意する必要がある。

既存手法では CTC に基づく認識結果に対して外部の言語モデルを用いたデコードを行うことによって、文全体の構文的一貫性や語彙的整合性といった言語的側面を補完していた。これに対して、本研究では、言語的側面を、視覚特徴と同時にエンコーダで学習可能であること、そしてこれが HTR の認識精度向上へ寄与することがわかった。以上より、本研究は HTR における言語情報統合の観点から、エンコーダに言語的表現力を獲得させる設計の有効性を示すとともに、この考え方が他の HTR モデルにも展開できる可能性を示唆した。

今後は、データ拡張等を用いてデータセットの数を増やした実験、ならびに外部に LLM を使用した場合との推論速度を含む総合的な評価を行い、提案手法の汎用性と実運用上の有効性を検証する。

謝辞

本研究をここまで進め、無事にまとめることができましたのは、多くの方々の支えがあってこそであり、深く感謝申し上げます。とりわけ、本研究の遂行に際し、専門的な知見に基づくご指導を賜っただけでなく、研究に取り組む姿勢や考え方、困難に直面した際の向き合い方に至るまで、常に寄り添いながら根気強く導いてくださいました大阪工業大学ロボティクス&デザイン工学部システムデザイン工学科 知能情報処理研究室の瀬尾昌孝准教授に、心より御礼申し上げます。日々のご指導と温かいお言葉の一つひとつが、研究を続ける上で大きな支えとなりました。

また、日々の研究活動の中で、温かく声をかけてくださり、的確な助言や励ましを与えてくださった知能情報処理研究室の大学院生である岡本聖也先輩、小林拓実先輩、上野遥平先輩、芥切優輔先輩、向康汰先輩、和住海利先輩に深く感謝いたします。皆様との議論や何気ない会話の一つ一つが、本研究を進める上で大きな支えとなりました。あわせて、同じ研究室で切磋琢磨し、ほぼ毎日、ニューラルネットワークについて理解が浅く深い理解ができていないところから辛いときも実験で結果が出てないときも14階で研究生活を送りあいともに成長しあった隣の席の髙橋清彌君に心より感謝申しあげます。また、ともにゼミで知識を出しあい成長させてくださった同期にも感謝いたします。

最後に、これまでの学生生活を通して常に支え続け、どんな時も見守ってくれた両親に、言葉では言い尽くせないほどの感謝の気持ちを伝えたいと思います。本研究は、多くの方々の支えの上に成り立っているものであり、ここに改めて深く感謝の意を表します。

参考文献

[1] M. Li et al.,“Trocr: Transformer-based optical character recognition with pre-trained models,” *AAAI*, 2023.

[2] J. Puigcerver et al., “Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition?,” Proc. ICDAR,2017.

[3] B. Shi et al., “An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition,” IEEE TPAMI, 2015.

[4] A. Vaswani et al., “Attention Is All You Need,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[5] D. Altinok et al., “Boosting CTC-Based ASR Using LLM-Based Intermediate Loss Regularization,” arXiv:2506.22846, 2025.

[6] M. Fujitake et al., “DTrOCR: Decoder-Only Transformer for Optical Character Recognition,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV),* pp. 8025-8035, 2024.

[7] P. B. Ströbel et al., “The Adaptability of a Transformer-Based OCR Model for Historical Documents,” in Document Analysis and Recognition - ICDAR 2023 Workshops, Lecture Notes in Computer Science, vol. 14193, pp. 34-48, doi: 10.1007/978-3-031-41498-5_3. ,2023

[8] D. Bautista et al., “Scene Text Recognition with Permuted Autoregressive Sequence Models,” in Computer Vision - ECCV 2022,

[9] U.-V. Marti et al., “The IAM-database: an English sentence database for offline handwriting recognition,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 5, pp. 39-46, 2002

[10] OpenSLR, “IAM Aachen splits,” SLR56, n.d.(accessed 2026-01-10)

[11] A. Grattafiori et al., “The Llama 3 Herd of Models,” arXiv:2407.21783, 2024