一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
PRMU2022-112,IBISML2022-119(2023-03)

Eye-contact Transformer: 骨格系列とシーン特徴による遠方歩行者のアイコンタクト検出

畑　隆聖†　出口　大輔†　平山　高嗣††,† 川西　康友†††,†村瀬　洋†

† 名古屋大学 大学院 情報学研究科　〒464–8601 愛知県名古屋市千種区不老町
†† 人間環境大学 環境科学部 環境データサイエンス学科　〒444–3505 愛知県岡崎市本宿町上三本松 6–2
††† 理化学研究所ガーディアンロボットプロジェクト　〒619–0088 京都府相楽郡精華町光台 2–2–2
E-mail: †hatar@vislab.is.i.nagoya-u.ac.jp

あらまし　車両の運転において，歩行者からのアイコンタクトは自車への気付きを判断する重要な要素である．従来のアイコンタクト検出手法の多くは眼球計測に基づく直接的な視線推定に依存しており，道路環境のように車両と歩行者の距離が離れるような場合は視線推定が困難である．一方で歩行者の骨格系列を用いた手法も提案されており，高精度なアイコンタクト検出を実現している．しかし，これらはいずれも歩行者のみに着目しており，他車両の存在等の周辺環境情報を考慮できていない．本報告では，特徴間の関係性を捉える Transformer ベースのモデルである Eye-contact Transformer を構築し，歩行者だけでなく周辺環境情報も加味することで，より高精度なアイコンタクト検出を実現する．車載カメラ画像を用いた実験により，その有効性を確認した．

キーワード　アイコンタクト検出，人物骨格系列，シーンコンテキスト，Transformer，self-attention

1. はじめに

車両の運転において，歩行者が自車両の存在に気づいているかどうかの判断は危険予測の観点から特に重要である．この気づきを判断するための重要な要素として，歩行者が自車両を見ているかどうか，すなわちアイコンタクトの有無がある．このような背景から，歩行者のより詳細な行動予測や周囲状況を詳細に考慮した自動運転の経路計画などの応用を見据え，アイコンタクトの有無を正確に認識する技術の実現が期待されている．アイコンタクトは，お互いに視線を交わす意味で使用される場合が多いが，本稿では歩行者が自車両を見る意味で使用する．

これまでに，アイコンタクト検出と関連が深いタスクである視線推定に関して多くの研究がなされてきた [1]〜[6] .

一例として，眼球を直接計測することで視線を推定する研究がある [1]〜[3]．これらは，赤外光照射装置と赤外線カメラを組み合わせた角膜反射法（PCCR）を用いており，高精度な視線推定が可能な手法である．しかし，前述のような特殊な装置が必要であり，また，赤外線の反射は個人差の影響を受けることから事前キャリブレーションが必要である，といった応用上での問題が存在する．

一方で，可視光画像から視線推定やアイコンタクト検出をする研究も存在する [4]〜[6] . Baltrusaitis らは顔画像から顔のランドマークを検出し，それを元に視線を推定する OpenFace を提案している [4] . この手法は両眼周辺のランドマークを正確に検出する必要があり，顔の向きや撮影距離に制限がある．また Zhang らは，顔画像から CNN で直接アイコンタクト検出をする手法を提案している [6] . しかし，学習段階で OpenFace を用いており，学習に用いる画像は顔のランドマークが鮮明に取れるものに限られている．そのため，離れた位置で撮影された解像度の低い歩行者に対して性能が低下することが予想される．また，Smith らは顔画像から目の周辺の領域だけをマスキングし，目の外観からアイコンタクトの有無を推定する手法を提案している [5] . しかしながら，使用したデータセットは頭の位置が器具で固定されており，顔向きが yaw 方向に −30°〜30°，pitch 方向は一定という条件下で撮影されたものであるため，横向きや後ろ向きの顔に対応可能かどうかは不明である．加えて，画像の背景は無地であるという制約もある．

図 1: 骨格系列での表現

以上のように，従来の手法の多くは，至近距離で撮影した高解像度の画像を要する，撮影環境に制約がある，被写体や状況毎にキャリブレーションが必要である，といった制約が存在する．そのため，対象人物に非常に近い位置にカメラを配置する必要があり，道路環境のように車両と歩行者の距離が離れるような状況においては，これらの手法による視線推定は不可能である．よって，遠方で解像度が低くなる歩行者のアイコンタクト検出のためには，視線推定に依らない手法が必要である．

(i) 対象歩行者領域のみ

(ii) 周辺情報込み

図 2: 周辺情報の有無によってアイコンタクトの判断が変わる可能性がある場合

これに対して，歩行者の 2 次元骨格情報を用いたアイコンタクト検出手法も提案されている [7], [8]．これらは人が普段車両を運転する際，視線推定が困難な歩行者に対しても顔向きや姿勢の時間変化を加味してアイコンタクトの有無を判断しているという考えに基づく手法である．Belkada らは，単一フレームの 2 次元骨格情報を用いたアイコンタクト検出手法を提案している [7]．また，我々は複数フレームの**骨格系列**を利用したアイコンタクト検出手法を提案している [8]．図 1 に歩行者の見た目を骨格系列で表現した例を示す．図 1(ii) の赤枠で示したフレームを見ると，アイコンタクト検出の重要な要素である顔向きや姿勢のねじれを骨格が表現していることがわかる．加えて，赤枠のフレーム単体だけでなく，それ以前のフレームも含めて見ると振り向きなどの動きも見て取れる．骨格系列を用いることで，直接的な視線推定が困難な遠方歩行者に対しても高精度なアイコンタクト検出を達成している．

しかしながら，これらは歩行者の見た目だけに着目した局所的な手法であり，歩行者とその周辺環境との関係性を考慮していない．実際の交通環境においては，対象歩行者と自車両以外にも，他車両や他歩行者，横断歩道や信号などが存在する．図 2 の (i) と (ii) の赤枠の歩行者を比較すると，対象歩行者の見た目のみの (i) ではこちらを見ていると判断できるが，周辺環境を踏まえた (ii) では右側の歩行者もしくは白い車を見ていると判断が変わる可能性がある．すなわち，対象歩行者のみでなく，他車両など周囲の物体の存在や，それらと対象歩行者との位置関係といった周辺環境情報（=**シーンコンテキスト**）も，アイコンタクト検出に影響する重要な要素と言える．

以上を踏まえ，本報告では歩行者の骨格系列とシーンコンテキストを利用したアイコンタクト検出手法である，Eye-contact Transformer（EyeT）を提案する．

2. Eye-contact Transformer

本論文では，骨格系列及びシーンコンテキストを利用することで遠方歩行者のアイコンタクトを検出する **Eye-contact Transformer（EyeT）**を提案する．EyeT は Transformer Encoder を持ち，その self-attention 層によって特徴量間の関係性を捉えることができる．これにより，歩行者の特徴（骨格系列）とシーンコンテキストの関係性を捉える．

EyeT の入力は，図 3 のように歩行者の 2 次元骨格系列と歩行者の BBox（対象歩行者のシーン内での位置や大きさを表現する外接矩形），シーン画像の 3 種類である．EyeT ではこれら 3 種類の異なる入力を同一形状の**トークン（ベクトル）**に変換することで，同じ Transformer の枠組みで扱うことを可能にしている．このために，EyeT には次の 2 つの工夫を施している．

（1） 異なる種類のトークンを同一の Transformer Encoder に入力するために，モデルがトークンの種類を区別可能になるよう，BERT [9] に倣い **Segment Embedding** を行なう．

（2） BBox をトークン化する際，少数のベクトルの加重和で高い分解能の位置表現を実現する **soft-label 表現**を行なう．

2.1 EyeT の処理手順

EyeT は，対象歩行者の T フレーム分の骨格系列 $\mathbf{x}_{\mathrm{kp}}$，T フレーム目の BBox 情報 $\mathbf{x}_{\mathrm{bbox}}$，$T$ フレーム目のシーン画像 $\mathbf{x}_{\mathrm{img}}$ を，各々対応する**バックボーン**で特徴量に変換し，Transformer を通して歩行者の T フレーム目に関するアイコンタクトの有無の推定結果 e_{pre} を次式のように出力する．

$$e_{\mathrm{pre}} = \mathrm{EyeT}(\mathrm{PE}(\mathbf{x}_{\mathrm{kp}}),\ \mathrm{B2V}(\mathbf{x}_{\mathrm{bbox}}),\ \mathrm{SE}(\mathbf{x}_{\mathrm{img}})) \tag{1}$$

図 3 に示すように，EyeT は 3 つのバックボーン（PE：Pedestrian Encoder，B2V：BBox2Vec，SE：Scene Encoder）による入力層と，埋め込み層，Transformer Encoder，MLP Head で構成される．EyeT の処理手順を以下に記す．また，以降の数式における変数の添字は，下付き文字は特徴量の種類，上付き文字はモデルの層番号をそれぞれ表す．

（1） **入力層**において，3 つの入力 $\mathbf{x}_{\mathrm{kp}}$，$\mathbf{x}_{\mathrm{bbox}}$，$\mathbf{x}_{\mathrm{img}}$ を**バックボーン**に通し，歩行者特徴量 $\mathbf{z}_{\mathrm{ped}}$，歩行者の BBox 特徴量 $\mathbf{z}_{\mathrm{bbox}}$，シーン画像特徴量 $\mathbf{z}_{\mathrm{img}}$ を以下のようにそれぞれ取得する．

$$\mathbf{z}_{\mathrm{ped}} = \mathrm{PE}\left(\mathbf{x}_{\mathrm{kp}}\right),\ \mathbf{z}_{\mathrm{ped}} \in \mathbb{R}^{1\times 768} \tag{2}$$

$$\mathbf{z}_{\mathrm{bbox}} = \mathrm{B2V}\left(\mathbf{x}_{\mathrm{bbox}}\right),\ \mathbf{z}_{\mathrm{bbox}} \in \mathbb{R}^{4\times 768} \tag{3}$$

$$\mathbf{z}_{\mathrm{img}} = \mathrm{SE}\left(\mathbf{x}_{\mathrm{img}}\right),\ \mathbf{z}_{\mathrm{img}} \in \mathbb{R}^{196\times 768} \tag{4}$$

（2） **埋め込み層**において CLS トークン $\mathbf{z}_{\mathrm{cls}}$ を生成し，$\mathbf{z}_{\mathrm{cls}}$ と，$\mathbf{z}_{\mathrm{ped}}$，$\mathbf{z}_{\mathrm{bbox}}$，$\mathbf{z}_{\mathrm{img}}$ を正規化層（LN：Layer Normalization [10]）に通して得た特徴量を連結した後，Position Embedding $\mathbf{E}_{\mathrm{pos}}$ 及び Segment Embedding $\mathbf{E}_{\mathrm{seg}}$ を付加する．これにより，Transformer Encoder の最初の層への入力テンソル $\mathbf{z}^{(0)}$ を取得する．

$$\mathbf{z}_{\mathrm{input}} = [\mathbf{z}_{\mathrm{cls}}, \mathrm{LN}(\mathbf{z}_{\mathrm{ped}}), \mathrm{LN}(\mathbf{z}_{\mathrm{bbox}}), \mathrm{LN}(\mathbf{z}_{\mathrm{img}})] \tag{5}$$

$$\mathbf{z}^{(0)} = \mathbf{z}_{\mathrm{input}} + \mathbf{E}_{\mathrm{pos}} + \mathbf{E}_{\mathrm{seg}},\quad \mathbf{E}_{\mathrm{pos}} \in \mathbb{R}^{(1+1+4+196)\times 768},\ \mathbf{E}_{\mathrm{seg}} \in \mathbb{R}^{(1+1+4+196)\times 768} \tag{6}$$

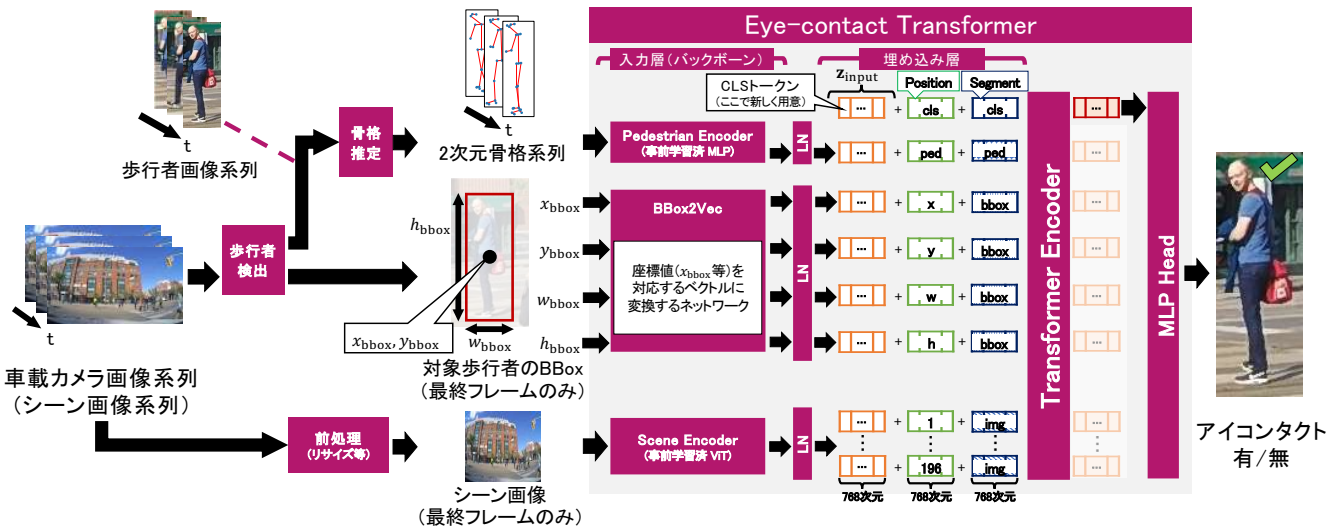


図 3: 提案手法の処理手順

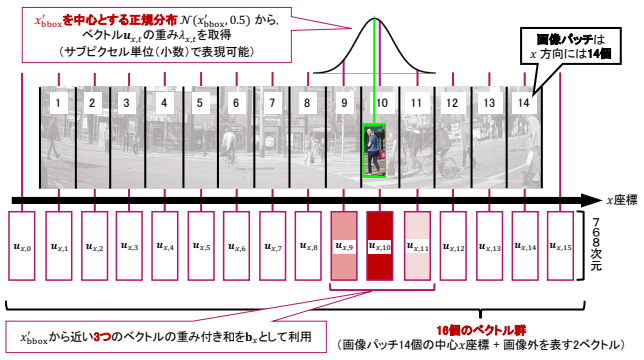


図 4: BBox2Vec の soft-label 表現

（3） $\mathbf{z}^0$ を **Transformer Encoder** に入力し，self-attention に基づいて各トークンを更新する．第 l 層の更新後のテンソルを $\mathbf{z}^{(l)}$ とし，次式により求める．

$$\widetilde{\mathbf{z}}^{(l)} = \mathrm{MSA}\big(\mathrm{LN}(\mathbf{z}^{(l-1)})\big) + \mathbf{z}^{(l-1)},\ l = 1, ..., L \tag{7}$$

$$\mathbf{z}^{(l)} = [\mathrm{MLP}\big(\mathrm{LN}(\widetilde{\mathbf{z}}^{(l)}_{\mathrm{cls}})\big), \mathrm{MLP}\big(\mathrm{LN}(\widetilde{\mathbf{z}}^{(l)}_{\mathrm{ped}})\big), \mathrm{MLP}\big(\mathrm{LN}(\widetilde{\mathbf{z}}^{(l)}_{\mathrm{bbox}})\big), \mathrm{MLP}\big(\mathrm{LN}(\widetilde{\mathbf{z}}^{(l)}_{\mathrm{img}})\big)] + \widetilde{\mathbf{z}}^{(l)},$$

$$\widetilde{\mathbf{z}}^{(l)} = [\widetilde{\mathbf{z}}^{(l)}_{\mathrm{cls}}, \widetilde{\mathbf{z}}^{(l)}_{\mathrm{ped}}, \widetilde{\mathbf{z}}^{(l)}_{\mathrm{bbox}}, \widetilde{\mathbf{z}}^{(l)}_{\mathrm{img}}],\ l = 1, ..., L \tag{8}$$

式 (7)，(8) の，MSA は multiheaded self-attention を表す．MSA によって得た $\widetilde{\mathbf{z}}^{(l)}$ を，MLP で非線形変換する．

（4） Transformer Encoder で更新されたトークンの特徴量 $\mathbf{z}^{(l)}$ のうち，CLS トークンに対応するベクトル $\mathbf{z}^{(L)}_{\mathrm{cls}}$ を，**MLP Head** に入力する．そして，T フレーム目に対する歩行者のアイコンタクトの有無を 2 クラス分類する．

$$e_{\mathrm{pre}} = \mathrm{MLP}_{\mathrm{Head}}(\mathbf{z}^{(L)}_{\mathrm{cls}}) \tag{9}$$

アイコンタクトの有無の予測結果 e_{pre} と真値 e_{th} のクロスエントロピー損失を求め，誤差逆伝播によって EyeT 全体の学習を行なう．

以下，EyeT（式 (1)）を構成する式 (2)〜(9) の実装を述べる．

2.2 入力層（バックボーン）

式 (2) の Pedestrian Encoder にはアイコンタクト検出タスクで事前学習した MLP モデルを用いる（3.2.1 項に後述）．また式 (4) の Scene Encoder は，Hugging Face の Transformers [11] に公開されている ViT 事前学習済みモデル[1]（以後，$\mathrm{ViT}_{\mathrm{hf}}$）を用いて実装する．

式 (3) の BBox2Vec は，歩行者の T フレーム目の BBox 情報 $\mathbf{x}_{\mathrm{bbox}} = [x_{\mathrm{bbox}}, y_{\mathrm{bbox}}, w_{\mathrm{bbox}}, h_{\mathrm{bbox}}]$ を入力とし，それぞれをベクトル表現に変換するネットワークである．BBox のベクトル表現については，LayoutLM [12] 等でも行なわれているが，従来方法は各座標を表すベクトル群を用意し，入力座標値に対応するベクトルを一つ出力する one-hot-label 表現である．各ベクトルは学習可能なパラメータベクトルであり，誤差逆伝播によって一から学習される．故に，ベクトルを増やすほど座標の表現能力が向上するが，それに伴い学習データが大量に必要となるデメリットがある．そこで BBox2Vec は，x，y，w，h それぞれについて，位置空間（サイズ空間）を表すためのベクトル群を持ち，その加重和によって入力値に対応するベクトルを表現する **soft-label 表現**を行なう．

ここでは，x を例に BBox2Vec の処理について説明する．本報告では，縦 1,080 画素，横 1,920 画素のシーン画像を扱う．シーン画像を Scene Encoder（ViT）に入力する際に x 方向は 14 個のパッチに分割される．BBox2Vec は，図 4 に示すように，**各シーン画像パッチの中心 x 座標と，両端の画像外を表現する 768 次元のベクトル**を 16（14+2）個持つ（$\mathbf{u}_x = [\mathbf{u}_{x,0}, \mathbf{u}_{x,1}, ..., \mathbf{u}_{x,15}]$）．これらのベクトルは，学習可能なパラメータベクトルである．

ここで，歩行者のサブピクセル単位の x 座標 x_{bbox} が入力されると，BBox2Vec は次のように，x_{bbox} に対応するベクトル $\mathbf{b}_x$ を出力する．

（1） $[0, 1920]$ の範囲の実数値 x_{bbox} を，$[0, 14]$ に正規化．

$$x'_{\mathrm{bbox}} = x_{\mathrm{bbox}} \times \frac{14}{1920} \tag{10}$$

（注1）：https://huggingface.co/google/vit-base-patch16-224-in21k

（2） x'_{bbox} から，関数 f（正規分布 $\mathcal{N}(x'_{\mathrm{bbox}}, 0.5)$）を作成．

$$f(x) = \frac{1}{\sqrt{2\pi} \times 0.5} \exp(-\frac{(x - x'_{\mathrm{bbox}})^2}{2 \times 0.5^2}) \tag{11}$$

（3） 関数 f を用いて，$\mathbf{u}_x$ の各ベクトルの重み $\boldsymbol{\lambda}_x = [\lambda_{x,0}, ..., \lambda_{x,15}]$ を得る．

$$\lambda_{x,t} = f(t - 0.5),\ t = 0, 1, ..., 15 \tag{12}$$

（4） $\boldsymbol{\lambda}_x$ のうち，最大値 $\lambda_{x,\hat{t}}$ を挟む **3 つ**の要素 $\lambda_{x,\hat{t}-1}$, $\lambda_{x,\hat{t}}$, $\lambda_{x,\hat{t}+1}$ を選択し，それらの合計が 1 になるように正規化する（図 4 参照）．それ以外は全て 0 にする．これにより，$\mathbf{u}_x$ の各ベクトルの，正規化された重み $\boldsymbol{\lambda}'_x = [\lambda'_{x,0}, \lambda'_{x,1}, ..., \lambda'_{x,15}]$ を得る．

（5） $\boldsymbol{\lambda}'_x$ を係数とするベクトル $\mathbf{u}_x$ の加重和で $\mathbf{b}_x$ を出力する．

$$\mathbf{b}_x = \boldsymbol{\lambda}'_x \times \mathbf{u}_x,\ \boldsymbol{\lambda}'_x \in \mathbb{R}^{1\times16},\ \mathbf{u}_x \in \mathbb{R}^{16\times768} \tag{13}$$

$\mathbf{b}_x$ が，$\mathbf{z}_{\mathrm{bbox}}$ の x 座標に関するトークンとなる．y，w，h についても処理は同様だが，式 (10) について，x と w では 1,920 で除するところを，y と h では 1,080 で除する点に注意されたい．

2.3 埋め込み層

EyeT の埋め込み層では，Position Embedding と共に，Segment Embedding を行なっているのが特徴である．図 3 に示すように，Segment Embedding は特徴量の種類ごとに異なるベクトル（CLS，Ped，BBox，Img の 4 種類）を用意して，各トークンの種類をモデルが識別できるようにしている．一方 Position Embedding は，各トークン別々のベクトルを用意して加算する．そのためモデルは，Position Embedding によって同一種類内の各トークンを区別することができる．Position Embedding，Segment Embedding は共に，学習可能なパラメータベクトルを用意し，誤差逆伝播によって一から学習される．ただし，Position Embedding のうち，CLS トークンと画像パッチに関するベクトルについては，$\mathrm{ViT}_{\mathrm{hf}}$ の重みを初期値として利用する．

2.4 Transformer Encoder

Transformer Encoder は，$\mathrm{ViT}_{\mathrm{hf}}$ の Transformer Encoder 12 層のうち，後ろ **3 層**を利用する．また各層，式 (8) に示すように，MSA で変換された各トークンを MLP で非線形変換する．この時，特徴量の種類毎に合わせた非線形変換を行なうために，トークン種類毎に異なる MLP を通す．トークンの種類毎に MLP を適用した後，それらを改めて連結する（式 (8)）．

2.5 MLP Head

MLP Head は，Dropout と線形層のシンプルな構造で実装する．

2.6 EyeT への入力の前処理

歩行者検出は，後述する PIE データセットに付与されている BBox の真値を利用する．骨格推定では，OpenPose [13] の BODY_25 モデルを使用し，尻の関節点が原点として座高（首と尻の関節点間の L2 距離）が人物間で揃うように正規化する．シーン画像の前処理は，$\mathrm{ViT}_{\mathrm{hf}}$ の Feature Extractor を利用する．

3. 評価実験

3.1 データセット

本実験では，Pedestrian Intention Estimation dataset（以後，PIE データセット）[14] をアイコンタクト検出タスク用に拡張した PIE+ データセットを作成して用いた．まず，PIE データセットは，157° の広角レンズを装備した車載カメラで撮影した約 900,000 枚の画像系列からなるデータセットである．PIE データセットには複数の歩行者が存在し，1,842 人の歩行者については，歩行者 ID，BBox，遮蔽率などの情報が，フレーム単位で延べ 738,970 枚にアノテーションされている（以後，歩行者データと呼ぶ）．

図 5: PIE+ データセットの歩行者データ例

表 1: 実験に用いた歩行者データの内訳

データの種類	データ拡張前のデータ数	データ拡張後のデータ数
Train	87,538	165,268
Validation	12,767	12,767
Test	65,102	65,102

この歩行者データ 738,970 枚に対して，PIE+ データセットでは，歩行者のアイコンタクトの有無を表す look_with_ratio ラベルを，1 歩行者データにつき 5 人のアノテータによって付与した．各アノテータは図 5 のような車載カメラ動画像を見て，0（アイコンタクト無）もしくは 1（アイコンタクト有）の 2 値でラベル付けし，それらをリスト化して look_with_ratio ラベルとしている（例：[0, 1, 1, 0, 1]（5 人中 3 人がアイコンタクト有，2 人がアイコンタクト無と判断した場合））．本実験では，look_with_ratio ラベルで 3 人以上が 1（アイコンタクト有）を付与したデータをアイコンタクト有，それ以外ではアイコンタクト無として，真値として用いた．

また本実験では，問題設定の簡単化のため，以下の条件を満たす歩行者データのみを用いた．

（1） 歩行者の遮蔽率が 25% 以下

（2） 歩行者領域で切り出した画像が縦 150 画素以上

（3） 歩行者領域で切り出した画像の矩形が，元の車載カメラ画像の枠（$1,920 \times 1,080$）を超えない

さらに，アイコンタクトの有無で Train データの数が同数となるようにデータ拡張を行なった．データの内訳を表 1 に示す．またこの際，$\frac{9}{10}$ の割合で次のような操作を施した．

（1） OpenPose による骨格推定結果の x, y 座標値に対して $N(0, 0.1)$ の正規分布に従うノイズを加える．

（2） $1,920 \times 1,080$ 画素のシーン画像を $1,918 \times 1,078$ 画素でランダムクロップした後，拡大して $1,920 \times 1,080$ 画素に戻

す操作を行ない，BBox の値も変更に併せて変動させる．

3.2 骨格系列とシーンコンテキストを用いる有効性の検証

骨格系列とシーンコンテキストを用いる効果の評価のために，以下の実験を行なった．

3.2.1 実 験 方 法

本実験では，表 2 に示す 10 手法の比較を行なった．⋆は ViT_{hf}，♠は事前学習済み 3D ResNet モデル[2]を表す．歩行者画像系列及び骨格系列は，10 フレーム分とした．また，提案 3 の Pedestrian Encoder の事前学習済みモデルには，提案 2 で最も高精度だった MLP を利用した．

評価指標には macro-F1 を用いた．macro-F1 は，真値がアイコンタクト有の歩行者データと無のデータそれぞれで求めた F 値の平均である．本実験では，各手法ランダムシードで 10 回ずつ試行し，各試行での macro-F1 の平均値 $\overline{F1}_{\text{Macro}}$ を比較した．

3.2.2 実 験 結 果

実験結果を表 2 に示す．骨格系列とシーンコンテキストを用いた手法 3 が最も $\overline{F1}_{\text{Macro}}$ が高く，骨格のみを利用した提案 1 よりも 0.1136，骨格系列を利用した提案 2 よりも 0.0181，$\overline{F1}_{\text{Macro}}$ が向上した．また図 6 に，骨格系列のみを用いた提案 2 ではアイコンタクトを誤検出したが，シーンコンテキストを加えた提案 3 では正しく検出した例を示す．

比較 1 は視線推定結果を用いたアイコンタクト検出手法だが，ほとんどの歩行者データに対して視線推定が不可能だった．

3.3 考 察

図 7 に，同一シーンにおける歩行者データ毎の，EyeT の Transformer Encoder 各層での **CLS トークンについてのシーン画像の Attention マップ**を示す．輝度が高い部分ほど，Attention がかかっている（CLS との関連性が高い）部分である．各画像，赤枠部分にアイコンタクト検出対象の歩行者が存在する．

図 7(ii) より，Transformer Encoder1 層目では信号機，前方の建物や空，他車両の一部に対して Attention がかかっていることが見て取れる．また 1 層目については，同一シーン内では，歩行者に依らず同様の Attention マップとなることがわかる．

続いて図 7(iii) の特に上段に注目すると，3 層目では対象歩行者が見ていると考えられる前方の車両の一部に Attention がかかっており，モデルがその存在を考慮していると期待できる．

一方で，図 7(iii) の下段を見ると，信号機付近に Attention がかかっている．このように，Transformer Encoder の 2 層目以降では，歩行者毎の位置や見た目（姿勢やその動き）の情報も踏まえ，歩行者毎に異なる Attention マップとなる．このことから，EyeT が歩行者の位置や見た目に応じて，シーン内の着目する部分を変えていることがわかる．

また，上記以外の例においても，道路や横断歩道，車，信号機など交通において重要な部分には，Attention がかかっていることを確認した．

4. む す び

本報告では，骨格系列とシーンコンテキストを用いたアイコンタクト検出手法である Eye-contact Transformer（EyeT）を提案した．EyeT は，シーン画像だけでなく骨格系列や BBox もトークン化して入力することで，異なる 3 種類の特徴量を同一の Transformer の枠組みで扱える．そして，self-attention でトークン間，すなわち歩行者と周辺環境との関係性を捉えることで，周辺環境情報を加味したアイコンタクト検出を行なう．PIE+ データセットを用いた評価実験を行い，最高精度を達成した．

今後の課題としては，シーン画像や BBox（位置情報）の時系列を扱うことの検証，実社会応用に向けたモデルの高速化などが挙げられる．

謝 辞

本報告の一部は JSPS 科研費（17H00745）による．

本報告内の実験は名古屋大学のスーパーコンピュータ「不老」を利用して実施した．

文 献

[1] C. Hennessey, B. Noureddin, and P. Lawrence, “A single camera eye-gaze tracking system with free head motion,” Proceedings of the Eye Tracking Research and Applications Symposium, vol.1, pp.87–94, Jan. 2006.

[2] D.H. Yoo and M.J. Chung, “A novel non-intrusive eye gaze estimation using cross-ratio under large head motion,” Computer Vision and Image Understanding, vol.98, no.1, pp.25–51, Apr. 2005.

[3] 江川晃一, 山本倫也, 長松　隆, “角膜反射法における視線計測可能ボリュームシミュレータの開発とマルチユーザ視線インタラクションシステムへの適用, ” 情報処理学会論文誌, vol.55, no.11, pp.2476–2486, Nov. 2014.

[4] T. Baltruaitis, A. Zadeh, Y.C. Lim, and L. Morency, “Openface 2.0: Facial behavior analysis toolkit,” Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, pp.59–66, May 2018.

[5] B.A. Smith, Q. Yin, S.K. Feiner, and S.K. Nayar, “Gaze locking: Passive eye contact detection for human-object interaction,” Proceedings of the ACM Symposium on User Interface Software and Technology, pp.271–280, Dec. 2013.

[6] X. Zhang, Y. Sugano, and A. Bulling, “Everyday eye contact detection using unsupervised gaze target discovery,” Proceedings of the ACM Symposium on User Interface Software and Technology, pp.193–203, Aug. 2017.

[7] Y. Belkada, L. Bertoni, R. Caristan, T. Mordan, and A. Alahi, “Do pedestrians pay attention? eye contact detection in the wild,” arXiv preprint arXiv:2112.04212, pp.1–10, Dec. 2021.

[8] R. Hata, D. Deguchi, T. Hirayama, Y. Kawanishi, and H. Murase, “Detection of distant eye-contact using spatio-temporal pedestrian skeletons,” Proceedings of the IEEE International Conference on Intelligent Transportation Systems, pp.2730–2737, Oct. 2022.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” Proceedings of the North American Chapter of the Association for Computational Linguistics, pp.4171–4186, Jun. 2019.

[10] J.L. Ba, J.R. Kiros, and G.E. Hinton, “Layer normalization,” arXiv preprint arXiv:1607.06450, pp.1–14, Jul. 2016.

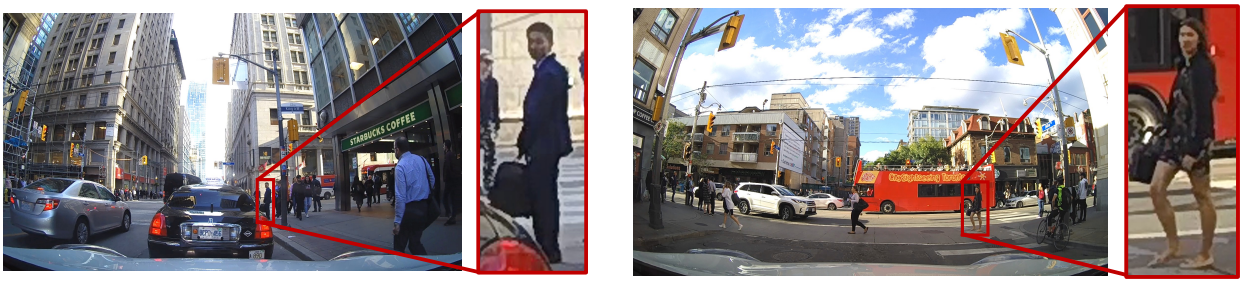
[11] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, “Transformers: State-of-the-art natural language processing,” Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp.38–45, Oct. 2020.

[12] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, “LayoutLM: Pre-training of text and layout for document image understanding,” Proceedings of the ACM International Conference on Knowledge

（注2）：https://pytorch.org/hub/facebookresearch_pytorchvideo_resnet/

表 2: 実験結果（10 回試行した平均と標準偏差で比較）

手法	モデル	入力	利用情報 骨格	動き	シーン	$\overline{F1}_{\text{Macro}} \pm$ 標準偏差
比較 1	OpenFace2.0 [4] + MLP	歩行者画像				視線推定が不可能
比較 2	3D ResNet [15]	歩行者画像				0.5216 ± 0.0093
比較 3	3D ResNet♠ [15]	歩行者画像				0.6358 ± 0.0159
比較 4	ViT★ [16]	歩行者画像				0.5143 ± 0.0077
比較 5	3D ResNet♠ [15]	歩行者画像系列		✓		0.6486 ± 0.0123
比較 6	ViT★ [16]	歩行者画像系列		✓		0.4870 ± 0.0042
比較 7	ViT★ [16]	歩行者画像系列 + シーン（画像 +BBox）		✓	✓	0.4296 ± 0.0540
提案 1	MLP [7]	骨格	✓			0.6033 ± 0.0025
提案 2	MLP [8]	骨格系列	✓	✓		0.6988 ± 0.0023
提案 3	**EyeT**	骨格系列 + シーン（画像 +BBox）	✓	✓	✓	**0.7169 ± 0.0040**



(i) 真値：アイコンタクト無　　(ii) 真値：アイコンタクト有

図 6: 提案 3（骨格系列 + シーンコンテキスト）で正しくアイコンタクト検出した例（提案 2（骨格系列のみ）では誤検出）

(i) 入力シーン画像　　(ii) Transformer Encoder 1 層目　　(iii) Transformer Encoder 3 層目

図 7: 同一シーンでの歩行者による CLS トークンに対する Attention マップの違い（対象歩行者を赤枠で表示）

Discovery and Data Mining, pp.1192–1200, Aug. 2022.

[13] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, “OpenPose: Realtime multi-person 2D pose estimation using part affinity fields,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.43, no.1, pp.172–186, Jan. 2021.

[14] A. Rasouli, I. Kotseruba, T. Kunic, and J. Tsotsos, “PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction,” Proceedings of the IEEE International Conference on Computer Vision, pp.6261–6270, Oct. 2019.

[15] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.6450–6459, Jun. 2018.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” Proceedings of the International Conference on Learning Representations, pp.1–21, Mar. 2022.