Eye-contact Transformer: シーンコンテクストを考慮した遠方歩行者のアイコンタクト検出

畑　隆聖[†a)]　出口　大輔[†]　平山　高嗣[†,††]　川西　康友[†,†††]
村瀬　洋[†]

Eye-Contact Transformer: Eye-Contact Detection of Distant Pedestrians Considering Scene Context

Ryusei HATA[†a)], Daisuke DEGUCHI[†], Takatsugu HIRAYAMA[†,††],
Yasutomo KAWANISHI[†,†††], and Hiroshi MURASE[†]

あらまし　歩行者が自車に気づいているかどうかの判断は，安全に車両を運転する上でとても重要である．このような自車への気付きの判断において，歩行者とのアイコンタクトは重要な役割を担っている．従来，このようなアイコンタクト検出には眼球計測により得られる視線推定結果が広く用いられており，道路環境のように歩行者と車両の距離が遠く歩行者の眼球を高解像度に計測できない場合は適用できない．そこで本論文では，歩行者の骨格系列と周囲環境の特徴を統合利用することにより，複雑な交通環境であっても歩行者とのアイコンタクトを精度良く検出可能な手法を提案する．具体的には，歩行者の骨格系列と周囲環境の特徴間の関係性をTransformer ベースのモデルで捉える Eye-contact Transformer を構築する．車載カメラ画像群に対してアイコンタクトのアノテーションを付与したデータセットを用いた実験により，提案手法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，シーンコンテクスト，Transformer，Self-Attention

1. まえがき

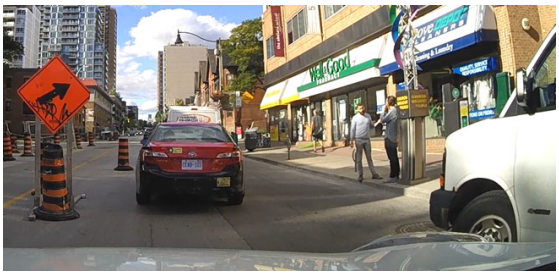
我々の日常的な車両での移動において，歩行者との安全なすれ違いや接触の回避などは運転者がよく遭遇するシーンである．このようなシーンでは慎重な判断が必要であり，歩行者が自車両に気づいているかどうかは安全な車両の運行において重要な要素となる．日々の運転行動を振り返ると，我々はこの気づきを判断するための重要な要素として歩行者とのアイコンタクトを確認していることに気づく．このような背景から，運転支援システムの高度化や自動運転車両の実現において，歩行者のより詳細な行動予測のための要素となるアイコンタクト検出技術の実現が期待されている．

図 1　骨格系列での表現

これまでに，顔画像を入力としてアイコンタクト検出を行う研究が幾つか提案されている [1], [2]．Zhang らは，顔画像を CNN に直接入力することでアイコンタクトを検出する手法を提案している [1]．しかし，顔画像の検出に OpenFace [3] を利用しており，対象者の顔を十分な解像度で撮影できない場合はアイコンタクト検出に必要となる目や口といった顔ランドマークが抽出できない．そのため，離れた位置で撮影された解像度の低い歩行者の場合は顔ランドマークの特徴を得

† 名古屋大学大学院情報学研究科，名古屋市
Nagoya Univ., Furo-cho, Chikusa-ku, Nagoya-shi, 464–8601 Japan
†† 人間環境大学環境科学部環境データサイエンス学科，岡崎市
Univ. of Human Environments, 6–2 Kami Sanbonmatsu, Motojuku-cho, Okazaki-shi, 444–3505 Japan
††† 理化学研究所，京都府
RIKEN, 2–2–2 Hikaridai, Seika-cho, Soraku-gun, Kyoto-fu, 619–0288 Japan
a) E-mail: hatar@vislab.is.i.nagoya-u.ac.jp
DOI:10.14923/transinfj.2023PDP0024

(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

ることができず，正しくアイコンタクト検出ができないという問題がある．一方，Smith らは顔画像から目の周辺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 [2]．しかしながら，横向きのように両目を確認できない場合はアイコンタクト検出できないという問題がある．これらの手法は，対象となる歩行者を至近距離で撮影した高解像度の画像が必要となる．そのため，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アイコンタクト検出は不可能である．

これに対して，運転者は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考慮してアイコンタクトの判断をしているというアイデアに基づき，歩行者の 2 次元骨格情報を入力としてアイコンタクトを検出する手法も提案されている．Belkada らは，単一フレームの 2 次元骨格情報を入力とするアイコンタクト検出手法を提案しており [4]，また我々は複数フレームの骨格系列を統合することで図 1 に示すような振り向き時のアイコンタクトなども高精度に検出が可能な手法を提案している [5]．

しかしながら，これらは対象歩行者から得られる情報のみ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，周囲に他の車両や歩行者，横断歩道や信号など様々な物体が存在する．例えば，図 2 の (i) と (ii) の赤枠の歩行者を比較すると，図 2 (i) の歩行者を見た場合はアイコンタクト有りと判断できるが，図 2 (ii) のように周辺環境を考慮すると自車両ではなく右側の歩行者若しくは白い車を見ていると判断が変わる可能性がある．このように，周囲に存在する物体，更にはそれらと対象歩行者との位置関係によってアイコンタクトの判断は変化する．このことから，アイコンタクト検出においては周辺環境情報（シーンコンテクスト）も重要な要素となる．

以上を踏まえ，本論文では歩行者の骨格系列とシーンコンテクストを利用することで複雑な交通環境下でも精度良くアイコンタクト検出が可能な手法である，Eye-contact Transformer（EyeT）を提案する．本論文の貢献は以下のとおりである(注1)．

（1） 歩行者の骨格系列情報とシーンコンテクストを Transformer の枠組みで統合する Eye-contact Transformer（EyeT）を提案する．これにより，振り向き等の歩行者の動きに加え，周囲に存在する車両等の物体との位置関係を考慮して複雑な交通環境下においても歩行者のアイコンタクト検出が可能なことを示す．

（2） 対象歩行者の矩形を少数のベクトルの加重和である soft-label 表現にすることにより，対象歩行者の矩形の表現能力を高く保ったまま Transformer に入力する手法を提案する．

（3） 車載カメラ画像に対してアイコンタクトの有無を付与したデータセットを構築し，パラメータを様々に変化させながら EyeT の性能を詳細に評価し，骨格系列情報とシーンコンテクストの組み合わせがアイコンタクト検出に有効であることを示す．

2. Eye-contact Transformer (EyeT)

本論文では，骨格系列及びシーンコンテク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 EyeT を提案する．EyeT は Transformer Encoder をもち，その Self-Attention 層によって歩行者の特徴（骨格系列）とシーンコンテクストの特徴量間での関係性を捉える．

EyeT の全体構成を図 3 に示す．図に示すように，

(注1)：本論文の内容は，追加実験により手法の詳細な評価をすることで文献 [6] の内容を発展的にまとめたものである．

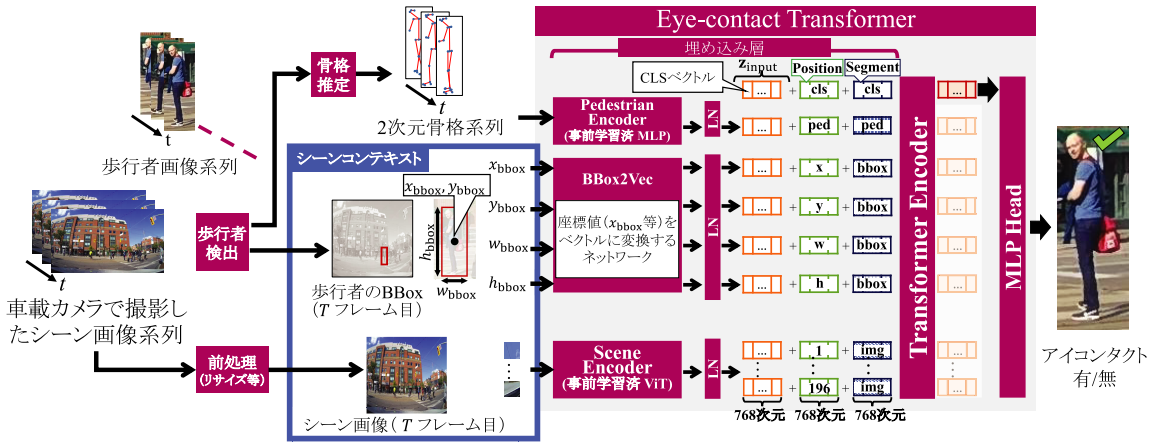


図 3　提案手法の処理手順

歩行者の 2 次元骨格系列，歩行者の BBox（対象歩行者のシーン内での外接矩形），シーン画像の 3 種類が EyeT の入力となる．EyeT では，以下の二つの工夫を加えることでこれら 3 種類の異なる入力を同一形状のベクトルに変換し，Transformer の入力として扱う．

（1） EyeT では，異なる種類の特徴ベクトルを同一の Transformer Encoder に入力し，それらの関係性を捉えることでアイコンタクト検出を行う．その際，入力された特徴ベクトルの種類（CLS ベクトル，骨格系列，BBox，シーン画像）を考慮して関係性を捉えられるよう特徴ベクトルの種類を表す Embedding を加える．具体的には，BERT [7] や MultiMAE [8] に倣って Segment Embedding を行う．詳細については **2.2.4** で説明する．また，CLS ベクトルは最終的なアイコンタクトの有無を出力する MLP Head の入力に用いられるベクトルであり，Transforer Encoder を通して歩行者の骨格系列，BBox，シーン画像の情報を集約した特徴ベクトルである．

（2） BBox の座標を少数のベクトルの加重和で表現することにより，必要なベクトル数を少なく保ちつつ高分解能な位置表現（soft-label 表現）を可能にする．soft-label 表現については，**2.2.3** で詳しく述べる．

以降の数式の説明においては，特徴量の種類を変数の右下の添字で区別し，モデルの層番号については変数の右上の添字として記す．

2.1　EyeT の処理手順

まず，T フレームからなる車載カメラ画像系列に対して OpenPose [9] 等の手法を適用し，歩行者の BBox 並びに骨格推定結果を得る．これにより，アイコンタクト判定の対象歩行者に対する骨格系列 $\boldsymbol{x}_{\mathrm{kp}}$ を取得し，提案手法の入力として用いる．そして，対象歩行者の T フレーム目における BBox 情報を $\boldsymbol{x}_{\mathrm{bbox}}$，$T$ フレーム目のシーン画像を $\boldsymbol{x}_{\mathrm{img}}$ とし，それらを図 3 に示す埋め込み層を通すことで特徴ベクトルに変換する．

$$\boldsymbol{z}_{\mathrm{ped}} = \mathrm{PE}(\boldsymbol{x}_{\mathrm{kp}}) \tag{1}$$

$$\boldsymbol{z}_{\mathrm{bbox}} = \mathrm{B2V}(\boldsymbol{x}_{\mathrm{bbox}}) \tag{2}$$

$$\boldsymbol{z}_{\mathrm{img}} = \mathrm{SE}(\boldsymbol{x}_{\mathrm{img}}) \tag{3}$$

ただし，$\boldsymbol{z}_{\mathrm{ped}} \in \mathbb{R}^{1\times 768}$, $\boldsymbol{z}_{\mathrm{bbox}} \in \mathbb{R}^{4\times 768}$, $\boldsymbol{z}_{\mathrm{img}} \in \mathbb{R}^{196\times 768}$ である．また，PE は骨格系列を特徴ベクトルに変換するバックボーン（Pedestrian Encoder）であり，B2V は対象歩行者の矩形を特徴ベクトルに変換するバックボーン（BBox2Vec）である．また，SE はシーン画像を変換するバックボーン（Scene Encoder）である．これら三つのバックボーンを通して得られる特徴量が，歩行者特徴量 $\boldsymbol{z}_{\mathrm{ped}}$，歩行者の BBox 特徴量 $\boldsymbol{z}_{\mathrm{bbox}}$，シーン画像特徴量 $\boldsymbol{z}_{\mathrm{img}}$ となる．次に，それらの特徴ベクトルを Transformer Encoder に入力し，MLP Head を通して対象歩行者の T フレーム目におけるアイコンタクトの有無の推定結果 e_{pre} を得る．

$$e_{\mathrm{pre}} = \mathrm{EyeT}(\boldsymbol{z}_{\mathrm{ped}}, \boldsymbol{z}_{\mathrm{bbox}}, \boldsymbol{z}_{\mathrm{img}}) \tag{4}$$

以降，具体的な EyeT の処理手順について詳しく述べる．まず，図 3 に示す埋め込み層において，CLS ベクトル $\boldsymbol{z}_{\mathrm{cls}}$ を新たに生成する．そして，式 (1)〜(3) で求めた $\boldsymbol{z}_{\mathrm{ped}}$，$\boldsymbol{z}_{\mathrm{bbox}}$，$\boldsymbol{z}_{\mathrm{img}}$ を正規化層（LN：Layer Normalization [10]）に通し，Position Embedding $\boldsymbol{E}_{\mathrm{pos}}$

及び Segment Embedding $\boldsymbol{E}_{\text{seg}}$ を付加する．これにより，Transformer Encoder の第 1 層への入力テンソル $\boldsymbol{z}^{(0)}$ を得る．

$$\boldsymbol{z}_{\text{input}} = \left[\boldsymbol{z}_{\text{cls}}, \text{LN}(\boldsymbol{z}_{\text{ped}}), \text{LN}(\boldsymbol{z}_{\text{bbox}}), \text{LN}(\boldsymbol{z}_{\text{img}})\right] \quad (5)$$

$$\boldsymbol{z}^{(0)} = \boldsymbol{z}_{\text{input}} + \boldsymbol{E}_{\text{pos}} + \boldsymbol{E}_{\text{seg}}, \quad (6)$$

ただし，$\boldsymbol{E}_{\text{pos}}$ と $\boldsymbol{E}_{\text{seg}}$ の次元は，CLS，$\boldsymbol{z}_{\text{ped}}$，$\boldsymbol{z}_{\text{bbox}}$，$\boldsymbol{z}_{\text{img}}$ それぞれに付加するため以下のとおりである．

$$\boldsymbol{E}_{\text{pos}} \in \mathbb{R}^{(1+1+4+196)\times 768},$$
$$\boldsymbol{E}_{\text{seg}} \in \mathbb{R}^{(1+1+4+196)\times 768},$$

$\boldsymbol{z}^{(0)}$ を Transformer Encoder に入力し，Self-Attention を通して各ベクトルを更新する．この処理を繰り返すことにより，Transformer の第 l 層の更新後に得られるテンソル $\boldsymbol{z}^{(l)}$ は次式により得られる．

$$\tilde{\boldsymbol{z}}^{(l)} = \text{MSA}\big(\text{LN}(\boldsymbol{z}^{(l-1)})\big) + \boldsymbol{z}^{(l-1)} \quad (7)$$

$$\boldsymbol{z}^{(l)} = \Big[\text{MLP}\big(\text{LN}(\tilde{\boldsymbol{z}}^{(l)}_{\text{cls}})\big), \text{MLP}\big(\text{LN}(\tilde{\boldsymbol{z}}^{(l)}_{\text{ped}})\big), \quad (8)$$
$$\text{MLP}\big(\text{LN}(\tilde{\boldsymbol{z}}^{(l)}_{\text{bbox}})\big), \text{MLP}\big(\text{LN}(\tilde{\boldsymbol{z}}^{(l)}_{\text{img}})\big)\Big] + \tilde{\boldsymbol{z}}^{(l)},$$

ただし，

$$\tilde{\boldsymbol{z}}^{(l)} = \left[\tilde{\boldsymbol{z}}^{(l)}_{\text{cls}}, \tilde{\boldsymbol{z}}^{(l)}_{\text{ped}}, \tilde{\boldsymbol{z}}^{(l)}_{\text{bbox}}, \tilde{\boldsymbol{z}}^{(l)}_{\text{img}}\right]$$

である．ここで，MSA は Multiheaded Self-Attention を表し，MSA により得られる $\tilde{\boldsymbol{z}}^{(l)}$ を MLP に入力する．Transformer Encoder が出力するベクトル $\boldsymbol{z}^{(l)}$ のうち，CLS ベクトルに対応する出力の $\boldsymbol{z}^{(L)}_{\text{cls}}$ を MLP Head に入力し，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 クラス分類の結果として得る．

$$e_{\text{pre}} = \text{MLP}_{\text{Head}}(\boldsymbol{z}^{(L)}_{\text{cls}}) \quad (9)$$

EyeT の学習においては，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 全体の学習を行う．

以下，EyeT を構成する埋め込み層の具体的な実装について述べる．

2.2 埋め込み層の実装方法

埋め込み層では，大きく性質の異なる 3 種類の入力を同一の Transformer で扱うために，バックボーンでそれぞれを 768 次元のベクトルに変換し，各ベクトルを区別するための情報（Position Embedding と Segment Embedding）を付与する．

2.2.1 Pedestrian Encoder (PE)

式 (1) 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 を用いる．これは，文献 [5] の比較実験として利用されている MLP モデルと同じ構造であり，後述する PIE+ データセットで事前学習を行ったものである．ここで，ある t フレームにおける関節点 i の 2 次元座標を $(x_{i,t}, y_{i,t})$，信頼度を $c_{i,t}$ とすると，MLP の入力である骨格系列 $\boldsymbol{x}_{\text{kp}}$ は，関節点座標と信頼度を T フレーム分並べた $(x_{1,1}, y_{1,1}, c_{1,1}, x_{2,1}, y_{2,1}, c_{2,1}, \ldots, x_{25,T}, y_{25,T}, c_{25,T})$ となる $3\times 25\times T$ 次元のベクトルである．出力は 768 次元ベクトルであり，事前学習を通してアイコンタクト検出の手がかりとなる歩行者の姿勢や動きを捉える特徴ベクトルを得る．

2.2.2 Scene Encoder (SE)

式 (3) の Scene Encoder (SE) は，ImageNet-21k [11] で事前学習済みの ViT モデル(注2)（以後，ViT_{hf}）を用いる．事前学習済みの ViT には，Hugging Face の Transformers [12] で公開されているものを利用する．ここで用いる ViT は，入力シーン画像を 196 個（14×14 個）の画像パッチに分割し，それぞれ ViT 内の Transformer を通して 768 次元のベクトルに変換する．

2.2.3 BBox2Vec (B2V)

式 (2) における BBox2Vec (B2V) は，歩行者の T フレーム目の矩形情報を表す $\boldsymbol{x}_{\text{bbox}} = (x_{\text{bbox}}, y_{\text{bbox}}, w_{\text{bbox}}, h_{\text{bbox}})^T$ 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．ここで，$(x_{\text{bbox}}, y_{\text{bbox}})$ は歩行者を囲う矩形の中心座標を表し，$(w_{\text{bbox}}, h_{\text{bbox}})$ は矩形の幅と高さを表す．従来の矩形のベクトル表現としては，各座標それぞれ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のみ値が 1 となる one-hot-label 表現で座標を表す方法が用いられてきた [13]．ここで各座標を表す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学習するのが一般的である．しかしこの方法では，ベクトル数を増やすほど座標の表現能力が向上するものの，それに伴って学習データが大量に必要となるというデメリットがある．そこで BBox2Vec は，x，y，w，h それぞれについて，位置を表すための少数のベクトルを用意し，その加重和によって入力座標を表現する soft-label 表現を用いる．

ここでは，x を例に BBox2Vec の処理について説明

（注2）：https://huggingface.co/google/vit-base-patch16-224-in21k

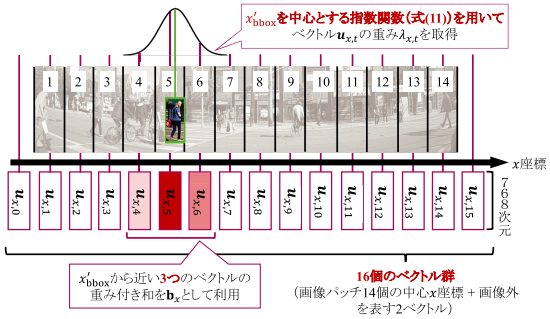


図 4　BBox2Vec の soft-label 表現

する．ただし，入力画像のサイズは 1,920 × 1,080 画素とする．2.2.2 で述べた Scene Encoder（ViT）の分割パッチ数を考慮し，BBox2Vec においては，ViT における各画像パッチの中心 x 座標に対応する 14 個のベクトルと，両画像端を表現する 2 個のベクトルを合わせた 16（14+2）個を基準ベクトルとして用いる．

$$\boldsymbol{u}_x = \left(\boldsymbol{u}_{x,0}, \boldsymbol{u}_{x,1}, \ldots, \boldsymbol{u}_{x,15}\right) \tag{10}$$

$\boldsymbol{u}_{x,0}, \ldots, \boldsymbol{u}_{x,15}$ それぞれのベクトルの次元は 768 であり，これらは学習可能なパラメータベクトルとする．

ここで，歩行者矩形の中心 x 座標を x_{bbox} とすると，BBox2Vec は次の手順により x_{bbox} に対応するベクトル $\boldsymbol{b}_x$ を出力する．まず，x_{bbox} の値の範囲が [0,14] となるように正規化したものを x'_{bbox}（$0 \le x'_{\text{bbox}} \le 14$）とする．次に，$x'_{\text{bbox}}$ に対応する各基準ベクトル $\boldsymbol{u}_x = \left(\boldsymbol{u}_{x,0}, \boldsymbol{u}_{x,1}, \ldots, \boldsymbol{u}_{x,15}\right)$ の各要素に対応する重み $\boldsymbol{\lambda}_x = (\lambda_{x,0}, \ldots, \lambda_{x,15})^T$ を次式により求める．

$$\lambda_{x,t} = \exp\left\{-\frac{(t - x'_{\text{bbox}} - 0.5)^2}{\sigma}\right\} \tag{11}$$

そして図 4 に示すように，$\boldsymbol{\lambda}_x$ の要素の中で値が最大となるものを $\lambda_{x,\hat{t}}$ とし，$\lambda_{x,\hat{t}-1}$，$\lambda_{x,\hat{t}}$，$\lambda_{x,\hat{t}+1}$ $(0 \le \hat{t} \le 15)$ の合計が 1 になるように正規化し，残りの要素を 0 にする．この手順により得られる正規化重みベクトルを $\boldsymbol{\lambda}'_x = (\lambda'_{x,0}, \lambda'_{x,1}, \ldots, \lambda'_{x,15})^T$ とする．そして，

$$\boldsymbol{b}_x = \boldsymbol{\lambda}'^T_x \boldsymbol{u}_x \tag{12}$$

により $\boldsymbol{z}_{\text{bbox}}$ の x 座標に関する特徴表現 $\boldsymbol{b}_x$ を得る．y，w，h に関する特徴表現についても上記と同様の手順により求める．

2.2.4　Position Embedding と Segment Embedding

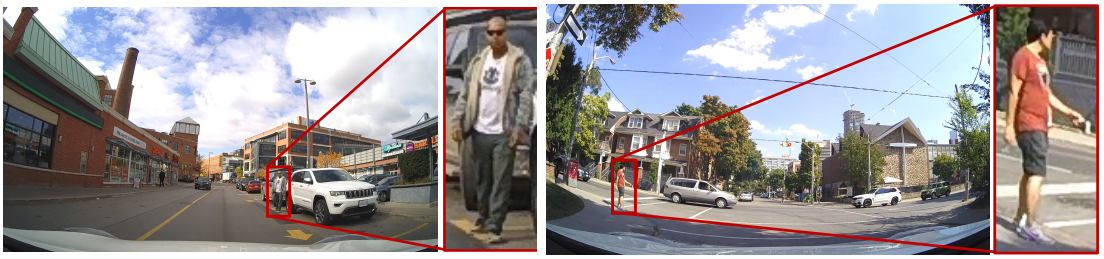
EyeT では，Position Embedding と Segment Embedding を組み合わせて用いる．図 3 に示すように，Segment Embedding は特徴量の種類ごとに異なるベクトル（CLS，Ped，BBox，Img の 4 種類）を用意し，各ベクトルの種類をモデルが識別できるようにする効果を狙ったものである．一方 Position Embedding は，入力特徴ベクトルごとに異なるベクトルを加算する．この Position Embedding により，同一種類であっても各入力特徴ベクトルを区別できるようにする．これらの Position Embedding と Segment Embedding はいずれも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学習する．

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 データセットと呼ぶ）[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実験に用いた．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り，1,842 人の歩行者に対して歩行者 ID，BBox，遮蔽率などの情報が延べ 738,970 枚のフレームに付与されている．PIE データセットは，歩行者横断意図予測タスクに主眼をおいて作成されたデータセットであり，PIE データセットには歩行者が自車両を見ているか否かを表す look ラベル（2 値ラベル）が付与されている．しかし，look ラベルの付与方法は明らかにされておらず，また，図 5 に示すような明らかなラベル誤りも多数含まれている．また，look ラベルは 0/1 の 2 値で付与されているため，アイコンタクトの有無の曖昧さ（例えば，「複数のアノテータ（今回は 5 人）の判断が一致する場合」や「アノテータによって判断が異なる場合」など）を加味した評価を行うことはできない．そこで PIE+ データセットは歩行者のアイコンタクト検出タスクに主眼をおき，上記の問題の解決を目的として作成したものである．

PIE+ データセットでは，前述の 738,970 枚の歩行者データに対して歩行者のアイコンタクトの有無を表す look_with_ratio ラベルを付与した．具体的には，歩行者の各フレームに対し，5 人のアノテータがアイコンタクトの有無を付与した．各アノテータは図 6 に示すような車載カメラ動画像を確認し，0（アイコンタクト



(i) look = 0（自車両を見ていないラベルが付与されえいる例）　(ii) look = 1（自車両を見ているラベルが付与されている例）

図 5　PIE データセットで look ラベルが明らかに誤っている例

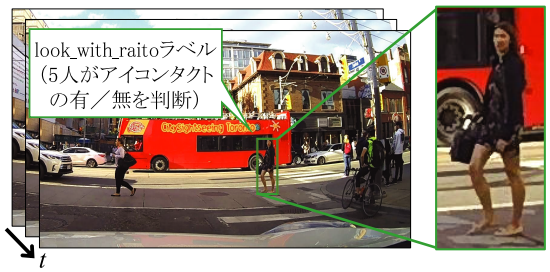


図 6　PIE+ データセットの歩行者データ例

無）若しくは 1（アイコンタクト有）の 2 値でラベル付けした．その際，アノテータには動画のシーン全体並びに前後のフレームを見ながら歩行者のアイコンタクトを判断するよう指示した．本実験では，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，学習と評価の真値として用いた．

本実験では，歩行者検出結果がアイコンタクト判定に与える影響を除くため，PIE データセットに付与されている BBox の真値を歩行者検出結果として用いた．また本実験では，OpenPose による骨格検出が安定して行える以下の歩行者画像を実験の対象として学習と評価に用いた．

（1）　歩行者の遮蔽率が 25% 以下

（2）　歩行者矩形の高さが縦 150 画素以上

また，アイコンタクトの有と無で学習データ数が同数となるようにデータ拡張を行った．データ拡張として，OpenPose による骨格推定結果の各座標値に対して $N(0, 0.1)$ のノイズ付与，入力画像のランダムクロップ＆リサイズ，を用いた．最終的に実験に用いたデータ数の内訳を表 1 に示す．

3.2　実 験 設 定

本実験では，表 2 に示す 10 種類の手法の比較を行った．

表 1　実験に用いた歩行者データ数の内訳

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

歩行者画像系列及び骨格系列の系列長は 10 フレームとし，提案 1 と提案 2 の Pedestrian Encoder の事前学習済みモデルには，**2.2.1** で述べたように文献 [5] の実験で利用している MLP を採用し，PIE+ データセットで事前学習を行ったモデル（比較 8）を用いた．

提案 1 は Transformer 構造をもたない提案手法であり，図 3 における Scene Encoder を ResNet50 [15], Transformer Encoder を MLP に置き換えたものである．提案 1 の ResNet50 モデルには ImageNet-1k [16] で事前学習したモデル(注3)を用いた．

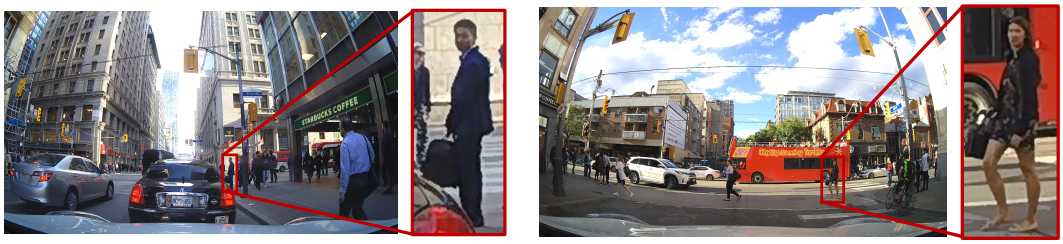
一方，比較 1 は OpenFace2.0 により得られる視線情報を MLP に入力してアイコンタクト検出する手法である．比較 2 と比較 4 は Kinetics 400 データセット [17] で事前学習した 3D ResNet モデル(注4)を用いてアイコンタクト検出をする方法であり，比較 3，比較 5，比較 6 は事前学習済み ViT モデル（**2.2.2 と同じ** $\mathrm{ViT_{hf}}$）を用いてアイコンタクト検出を行う手法である．比較 7 と比較 8 は骨格情報を用いてアイコンタクト検出を行う従来手法 [4], [5] である．

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とアイコンタクト無の歩行者データそれぞれで求めた F 値の平均により求める．本実験では，各手法に対してランダムなシードを与えて 10 回ずつ試行し，各試行での macro-F1 の平均値 $\overline{F1}_{\mathrm{Macro}}$ を評価に用いた．

（注3）：https://huggingface.co/docs/transformers/model_doc/resnet
（注4）：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	$\overline{F1}_{\mathrm{Macro}}$ ± 標準偏差
比較 1	OpenFace2.0+ MLP	歩行者画像				視線推定が不可能
比較 2	3D ResNet	歩行者画像				0.6358 ± 0.0159
比較 3	ViT	歩行者画像				0.5143 ± 0.0077
比較 4	3D ResNet	歩行者画像系列		✓		0.6486 ± 0.0123
比較 5	ViT	歩行者画像系列		✓		0.4870 ± 0.0042
比較 6	ViT	歩行者画像系列 + シーン（画像 +BBox）		✓	✓	0.4296 ± 0.0540
比較 7	MLP [4]	骨格	✓			0.6033 ± 0.0025
比較 8	MLP [5]	骨格系列	✓	✓		0.6988 ± 0.0023
提案 1	EyeT(ResNet50, MLP)	骨格系列 + シーン（画像 +BBox）	✓	✓	✓	0.6930 ± 0.0221
提案 2	**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 7　提案 2（骨格系列 + シーンコンテクスト）でアイコンタクトの有無を正しく判定した例（比較 8（骨格系列のみ）ではいずれも誤判定）

(i) 腕を曲げている（真値：アイコンタクト有）　(ii) 腕を曲げている（真値：アイコンタクト有）　(iii) 骨格の誤推定（真値：アイコンタクト無）　(iv) 特殊な姿勢（真値：アイコンタクト有）

図 8　EyeT でアイコンタクトの判定を誤った例

表 3　アイコンタクト検出結果の混同行列

(i) 比較 8（MLP [5]）

		予測結果 有	無
真値	有	1,919	635
	無	4,288	58,247

(ii) 提案 2（EyeT）

		予測結果 有	無
真値	有	1,747	806
	無	3,194	59,341

3.2.1　実験結果

各手法の比較結果を表 2 に示す．骨格系列とシーンコンテクストを用いた提案 2 が最も $\overline{F1}_{\mathrm{Macro}}$ が高く，骨格のみを利用する比較 7 よりも 0.1136，骨格系列を利用する比較 8 よりも 0.0181，$\overline{F1}_{\mathrm{Macro}}$ が向上することを確認した．表 3 は，骨格系列のみを用いる比較 8 と，シーンコンテクストを利用する提案 2 の混同行列を示している．表 3 より，シーンコンテクストを用いる提案 2 では，アイコンタクト無しに対する正解数が比較 8 よりも増えていることが見て取れる．図 7 は，骨格系列のみを用いた比較 8 ではアイコンタクトを誤検出したものの，シーンコンテクストを加えた提案 2 では正しく検出した例を示している．

従来から広く用いられている視線推定に基づくアイコンタクト検出手法である比較 1 は，ほぼ全ての歩行者で視線推定ができず，アイコンタクト検出に失敗し

た．また，Transformer 構造をもたない提案 1 は，シーンコンテクストを加えてはいるものの比較 8 よりも精度が低下することを確認した．

図 8 に，EyeT でアイコンタクト検出が失敗した歩行者データの例を示す．まず，図 8 (i)，(ii) のように，腕を曲げている歩行者が散見された．これは，スマートフォンを見ている人物の姿勢と近い．スマートフォンを見ている歩行者データは一定数存在するため，その影響でアイコンタクト検出に失敗したと考えられる．また，図 8 (iii) は，OpenPose による骨格推定を誤った例である．手前に重なった人物とまとめて一人として推定されており，正しい骨格情報が得られていない．更に図 8 (iv) のような屈んでいる場合など，学習データに少数しか含まれていない姿勢をしている歩行者に対するアイコンタクト検出誤りが見られた．

4. 考　　察

4.1 シーンコンテクストの効果

図 9 は，同一シーン内に存在する 2 人の歩行者それぞれに対してアイコンタクト検出を行い，EyeT の Transformer Encoder の 1 層目と 3 層目における CLS ベクトルに関連する Attention マップを示している．各画像中の赤枠部分がアイコンタクト検出対象の歩行者であり，図中の明るい領域ほど Attention の値が高い（CLS との関連性が高い）ことを表している．図 9 (ii) より，Transformer Encoder の 1 層目は信号機，前方の建物，空，他車両などに対して Attention が強く反応していることが見て取れる．また 1 層目から得られる Attention マップに関しては，対象となる歩行者による違いは見られないことがわかる．

一方，図 9 (iii) の上段に注目すると，3 層目では対象歩行者が注視していると考えられる前方車両の一部で Attention が高い値を示しており，EyeT でその存在がアイコンタクト検出に考慮されていることがわかる．次に図 9 (iii) の下段を見ると，信号機付近の Attention が高い値を示していることから，Transformer Encoder の 2 層目以降では歩行者の位置や姿勢等の情報も踏まえて Attention マップが計算されており，歩行者ごとに異なる結果となっている．このことから，EyeT が歩行者の位置や姿勢に応じて，シーン内の着目する部分を変化させていることを確認した．

4.2 BBox2Vec による矩形のベクトル表現の効果

BBox2Vec による矩形のベクトル表現の効果を確認するために，**3.1** と同様のデータセット並びに評価指標を用いて表 4 に示す五つの方法を比較した．

表 4 に示す「BBox 無し」は EyeT に BBox を入力せずにアイコンタクト検出する手法であり，歩行者骨格系列とシーン画像のみを入力としてアイコンタクト検出するものである．次に「one-hot-label」は，**2.2** で記述したように，用意したベクトルの数によって各座標値を表現できる分解能が変化する．本実験では，このことを踏まえ，表 4 に示す 3 種類の分解能で実験を行った．

最後の「soft-label」は，BBox の座標値 (x, y, w, h) それぞれに対応するベクトルを，ベクトルの重み付き和で表現する提案手法である．**2.2** で前述したように，少数の離散的なベクトルの組み合わせによって連続的な座標値を表現できるため，少数のベクトルであっても分解能を高めることができるものである．

各手法のアイコンタクト検出結果の F 値のマクロ平均に関して，10 回試行の平均値である $\overline{F1}_{\mathrm{Macro}}$ を表 4 に示す．表 4 より，soft-label 表現の場合が最も $\overline{F1}_{\mathrm{Macro}}$ が高く，BBox 情報を用いない場合と比較して 0.0074 向上した．また，one-hot-label の各手法は座標値の表現に用いるベクトル数が異なっており，14 個の場合は $\overline{F1}_{\mathrm{Macro}}$ は 0.7149 と提案手法の soft-label に近い値が得られたものの，分解能を上げるにつれて精度が低下し，シーン画像の解像度と同じ y 方向 1080，x 方向 1920 の分解能とした場合（画素単位で表現ベクトルを用意した場合）は $\overline{F1}_{\mathrm{Macro}}$ が 0.7087 と最も低い値となった．

以上の結果から，アイコンタクト検出対象歩行者の位置情報をコンテクストとして利用することが有効であることを確認し，また，提案する soft-label 表現が最も高い精度を示すことを確認した．一方，one-hot-label 表現で用いるベクトル数を増やすにつれて精度が低下した．これは，細かい表現が可能になる反面，ベクトル数の増加に伴って必要な学習データ数が増加し，今回の評価実験に用いたデータ数では不十分であった可能性が考えられる．

4.3 EyeT の Transformer Encoder の効果

EyeT で利用している Transformer Encoder の効果を確認するために，Transformer Encoder を異なるネットワークに置き換えた手法と比較した．具体的には，**3.1** と同様のデータセット並びに評価指標を用いて表 5 に示す三つの手法を比較した．

表 5 に示す Average Pooling は，z_{input}（CLS ベクトル及び三つのバックボーンを通して得られた合計 202

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

図 9　同一シーンでの歩行者による CLS ベクトルに対する Attention マップの違い（対象歩行者を赤枠で表示）

表 4　EyeT に入力する BBox のベクトル表現を変えた場合の実験結果（10 回試行した平均と標準偏差で比較）

BBox の表現方法	各座標値を表現するベクトルの数	分解能	$\overline{F1}_{\mathrm{Macro}}$ ± 標準偏差
BBox 無し	0	–	0.7095 ± 0.0029
one-hot-label	14	14	0.7149 ± 0.0037
one-hot-label	224	224	0.7143 ± 0.0042
one-hot-label	(x 座標，画像幅 w)：1920 (y 座標，画像高さ h)：1080	1920 1080	0.7087 ± 0.0021
soft-label	14+2	制限なし	**0.7169 ± 0.0040**

表 5　EyeT の Transformer Encoder を異なるネットワークに置き換えた場合の実験結果

手法	EyeT の Transformer Encoder 部分	$\overline{F1}_{\mathrm{Macro}}$± 標準偏差
Average Pooling	Average Pooling	0.6328 ± 0.0245
Concat	MLP	0.6955 ± 0.0131
EyeT(提案手法)	Transformer Encoder	**0.7169 ± 0.0040**

個の 768 次元ベクトル）を単純に平均して一つの 768 次元ベクトルとし，それを MLP Head を用いてアイコンタクト検出を行う手法である．一方 Concat は，z_{input} の 202 個のベクトルを全て連結して MLP に入力して 768 次元ベクトルに変換し，それを MLP Head に入力してアイコンタクト検出を行う手法である．

それぞれの手法でアイコンタクト検出に関する F 値のマクロ平均を求め，その試行を 10 回繰り返して求めた平均 $\overline{F1}_{\mathrm{Macro}}$ を表 5 に示す．Average Pooling や Concat と比べ，Transformer を用いる提案手法が最も高い精度が得られることを確認した．これは，Transformer がもつ Self-Attention 機構によって，シーンコンテクストと対象歩行者の特徴間で関係性をうまく捉えられるようになったためだと考えられる．

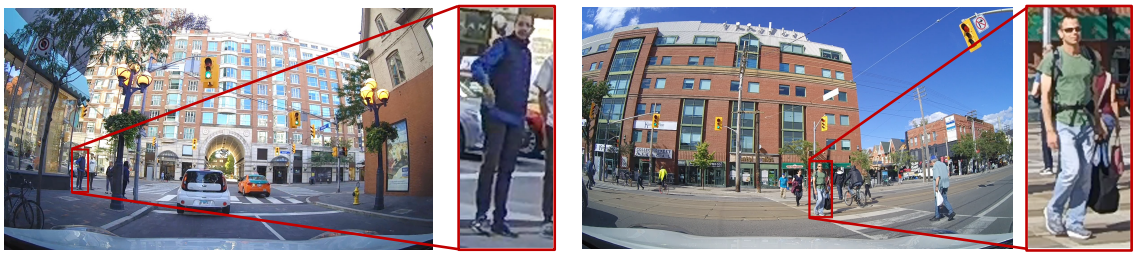
4.4　EyeT の Scene Encoder の効果

EyeT で利用している Scene Encoder の効果を確認するために，Scene Encoder を異なるネットワークに置き換えた手法と比較した．具体的には，**3.1** と同様のデータセット並びに評価指標を用いて表 6 に示す三つの手法を比較した．

Scene Encoder の替わりに ResNet50 を用いる手法では，ResNet50 の出力が ViT と同数の 14×14 個のパッチに分割された特徴量（ベクトル）となるようにしたものである．ここで ResNet50 のモデルには，評価実験で示した提案 1 と同様，ImageNet-1k [16] で事前学習したモデルを用いた．また SwinV2 においては，ViT の発展系である Swin Transformer [18] を ViT の替わりに用いたものであり ImageNet-1k で事前学習済みの SwinV2 モデル(注5) を用いた．

それぞれの手法でアイコンタクト検出に関する F 値のマクロ平均を求め，その試行を 10 回繰り返して求めた平均 $\overline{F1}_{\mathrm{Macro}}$ を表 6 に示す．ResNet50 を Scene Encoder の替わりに用いる手法と比べ，Transformer ベー

(注5)：https://huggingface.co/docs/transformers/model_doc/swinv2



(i) 5 人中 2 人がアイコンタクト有と判断　　(ii) 5 人中 3 人がアイコンタクト有と判断

図 10　EyeT がアイコンタクトの判定を誤り，アノテータの判断も分かれた例

表 6　EyeT の Scene Encoder を異なるネットワークに置き換えた場合の実験結果

Scene Encoder	$\overline{F1}_{\text{Macro}}\pm$ 標準偏差
ResNet50	0.6903 ± 0.0205
ViT（提案手法）	**0.7169 ± 0.0040**
SwinV2	0.7156 ± 0.0031

表 7　アイコンタクト有りと判断したアノテータの割合と EyeT の正解率

アノテータの判断	0/5	1/5	2/5	3/5	4/5	5/5
正解ラベル	無	無	無	有	有	有
データ数	60,520	1,211	804	699	843	1,025
正解率	96.6%	61.1%	49.3%	60.5%	67.9%	71.3%

スのモデルである ViT や SwinV2 を用いる手法が高い精度を示すことを確認した．交通シーンにおけるアイコンタクト検出においては，歩行者や他車両，歩道や信号等の様々な物体同士の相互関係を加味した判定が必要となる．Transformer では，Self-Attention 機構によってこのような特徴間の関係性，更には大域的な特徴を捉えることができるようになり，ViT や SwinV2 を用いる場合は ResNet のような CNN ベースの手法よりも高い性能を示したと考えられる．

4.5　複雑なシーンにおけるアイコンタクト検出

提案手法である EyeT におけるシーンコンテクスト利用の効果を確認するため，複雑なシーンでのアイコンタクト検出性能を調べる実験を行った．具体的には，複雑なシーンではアイコンタクト有無の判断がアノテータによって異なると考え，PIE+ データセット構築時に各アノテータがアイコンタクト有りと判断した割合と EyeT のアイコンタクト判定の正解率を調査した．表 7 は，アイコンタクト有りと判断したアノテータの割合と EyeT の正解率の関係を示している．表 7 中のアノテータの判断については，アノテータ 5 人中の何人がアイコンタクト有りと判断したかを示している．具体的には，0/5 の列は全てのアノテータがアイコンタクト無しと判断したデータであり，3/5 の列は 5 人中 3 人がアイコンタクト有りと判断したデータを示している．正解ラベルの無／有はアノテータの判断の多数決により付与した正解ラベルであり，EyeT の正解率の計算に用いたものである．また図 10 は，EyeT がアイコンタクトの判定を誤り，かつ，アノテータのアイコンタクトの判断も分かれた例を示している．図 10 並びに表 7 中の 0/5 と 5/5 の列より，アノテータが判断に迷わない（全員の判断が一致する）データでは EyeT の正解率が高く，2/5 の列のようにアノテータの判断が大きく分かれる難しいデータでは EyeT の正解率が低くなることを確認した．一方，1/5 や 4/5 の列のようにアノテータの判断の一致度が上がるにつれて，EyeT の正解率が高くなることも確認した．これらのことから，人によるアイコンタクトの判断が大きく分かれるような特に難しい状況への対処が今後必要であると考えられる．

5. むすび

本論文では，骨格系列とシーンコンテク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，シーン画像に加えてアイコンタクト検出対象歩行者の骨格系列と矩形情報もベクトル化して入力し，Transformer の枠組みで異なる 3 種類の特徴を扱う方法を提案した．そして，Transformer がもつ Self-Attention 機構により，歩行者と周辺環境との関係性を捉えながらアイコンタクト検出を行うことを可能とした．PIE+ データセットを用いた評価実験を行い，提案する EyeT がシーンコンテクストを考慮しない手法よりも高い精度を示すことを確認した．

今後の課題としては，動きを考慮したシーンコンテ

クストへの拡張，実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（23H03474）による．本論文内の実験には，名古屋大学のスーパーコンピュータ「不老」を利用した．

文　献

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

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

[3] T. Baltruaitis, A. Zadeh, Y.C. Lim, and L. Morency, “Openface 2.0: Facial behavior analysis toolkit,” Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition, pp.59–66, May 2018.

[4] 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.

[5] R. Hata, D. Deguchi, T. Hirayama, Y. Kawanishi, and H. Murase, “Detection of distant eye-contact using spatio-temporal pedestrian skeletons,” Proc. IEEE Int. Conf. Intelligent Transportation Systems, pp.2730–2737, Oct. 2022.

[6] 畑　隆聖，出口大輔，平山高嗣，川西康友，村瀬　洋，“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，”信学技報，PRMU2022-112, March 2023.

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

[8] R. Bachmann, D. Mizrahi, A. Atanov, and A. Zamir, “MultiMAE: Multi-modal multi-task masked autoencoders,” Proc. IEEE European Conf. Computer Vision, Springer-Verlag, Berlin, Heidelberg, pp.348–367, Oct. 2022.

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

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

[11] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, “ImageNet-21K Pretraining for the masses,” Proc. Neural Information Processing Systems, pp.1–9, Dec. 2021.

[12] 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,” Proc. Conf. Empirical Methods in Natural Language Processing: System Demonstrations, pp.38–45, Oct. 2020.

[13] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, “LayoutLM: Pre-training of text and layout for document image understanding,” Proc. ACM Int. Conf. Knowledge Discovery and Data Mining, pp.1192–1200, Aug. 2022.

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

[15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.770–778, June 2016.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.248–255, June 2009.

[17] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, “The kinetics human action video dataset,” arXiv:1705.06950 (doi:10.48550/arXiv.1705.06950), May 2017.

[18] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, “Swin Transformer V2: Scaling up capacity and resolution,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.11999–12009, June 2022.

（2023 年 5 月 26 日受付，9 月 29 日再受付，
11 月 30 日早期公開）

畑　　隆聖

令 3 名大・情・コンピュータ科卒．令 5 同大大学院情報学研究科博士前期課程了．修士（情報学）．同年株式会社 NTT ドコモ入社．

出口　大輔　（正員）

平 13 名大・工・情報卒．平 18 同大大学院情報科学研究科博士後期課程了．博士（情報科学）．平 16〜18 まで日本学術振興会特別研究員．平 18 名大大学院情報科学研究科研究員，平 18 名大大学院工学研究科研究員，平 20〜24 まで同大大学院情報科学研究科助教，平 24 より同大情報連携統括本部情報戦略室准教授，令 2 より同大大学院情報学研究科准教授．現在に至る．主に画像処理・パターン認識技術の開発とその ITS 及び医用応用に関する研究に従事．情報処理学会，IEEE 各会員．

平山　高嗣　（正員）

平 10 奈良高専・情報卒．平 12 金沢大・工・電気情報卒．平 17 阪大大学院基礎工学研究科博士後期課程了．博士（工学）．同年より京大大学院情報学研究科特任助教．平 23 より名大大学院情報科学研究科特任助教．平 24 より同助教．平 26 より同特任准教授．令 3 より人間環境大学環境科学部教授，現在に至る．顔画像認識，注視行動分析，視覚的注意の計算モデルに関する研究に従事．情報処理学会，IEEE，ACM 各会員．

川西　康友　（正員：シニア会員）

平 18 京大・工・情報卒．平 24 同大大学院情報学研究科博士後期課程了．博士（情報学）．平 24 同大学術情報メディアセンター特定研究員．平 26 名大未来社会創造機構特任助教．平 27 同大情報科学研究科助教．平 29 同大情報学研究科助教．令 2 同大情報学研究科講師．令 3 理化学研究所 ガーディアンロボットプロジェクト 感覚データ認識研究チーム チームリーダー．現在に至る．ロボットによる周囲環境認識及び，人物追跡・属性認識・行動認識などの人物画像処理に関する研究に従事．IEEE，画像電子学会各会員．

村瀬　　洋　（正員：フェロー）

昭 53 名大・工・電気卒．昭 55 同大大学院修士課程了．同年日本電信電話公社（現 NTT）入社．平 4 より 1 年間米国コロンビア大客員研究員．平 15 より名古屋大学大学院情報科学研究科教授．令 3 より同大名誉教授及び特任教授，現在に至る．文字・図形認識，コンピュータビジョン，マルチメディア認識の研究に従事．工博．IEEE フェロー，IAPR フェロー，情報処理学会フェロー．