プロトタイプ法 ProtoPFormer への人の知見の組み込みによる精度向上

落合 祐馬 †　平川 翼 †　山下 隆義 †　藤吉 弘亘 †
† 中部大学
E-mail: ochiai20031@mprg.chubu.ac.jp

1　はじめに

プロトタイプ法のモデルとは，機械学習モデルの一種でり，内部にプロトタイプと呼ばれるベクトルを保持する手法である．学習時にはプロトタイプの値をデータに適合するように最適化し，推論時にはこれらのプロトタイプベクトルとの類似度に基づいて判断を行う．プロトタイプベースのメリットは画像認識においてプロトタイプとの類似度を判断根拠として利用できる点にある．これにより，モデルが注目する領域をプロトタイプと位置から直接的に解釈可能となる．しかしながら，プロトタイプ法には課題も存在する．プロトタイプは学習によって得られるため，人間が期待する領域とは異なる不適切な領域に注目する場合がある．特にプロトタイプが対象外の背景領域や無関係な特徴に収束すると，判断根拠が不正確となり，分類精度の低下を招く．既存手法では，このような不適切な注目を修正する機構が備わっておらず，根本的な解決が困難であった．

本研究ではこれらの課題を解決するため，人の知見を損失関数に組み込む新たな手法を提案する．具体的には，人の知見を用いた損失関数を導入し，プロトタイプが適切な領域に収束するよう誘導する．これにより，モデルの判断根拠を改善しながら分類精度の向上を図る．

2　関連研究

本章では，プロトタイプベースの説明に加え，プロトタイプを用いた代表的な手法である ProtoPNet と ProtoPFormer について述べる．

2.1　プロトタイプベース

プロトタイプベースとは，機械学習においてプロトタイプと呼ばれる代表的な特徴パターンを参照点として意思決定を行う手法である．特に解釈可能性がもとめられる医療画像診断などの分野で重要な役割を持つ．

Prototypical Part Network（ProtoPNet）[8] や ProtoPFormer（Prototypical Part Transformer）[7] はこのプロトタイプを用いた代表的なモデルアーキテクチャである．ProtoPNet が Convolutional Neural Network (CNN) [1] をベースにしたモデルであり，ProtoPFormer が Vision Transformer (ViT) [3] を採用したモデルである．両モデルともプロトタイプによる局所的な注目の可視化を行うことができる．

2.2　ProtoPNet

CNN を用いたプロトタイプベースのモデルであり，プロトタイプは学習することで担当するクラスを表現するベクトルになる．画像分類を行う際プロトタイプを使用することでこのモデルは式 (1) で表される損失関数を用いて学習する．

$$\min_{P,w_{\mathrm{conv}}} \frac{1}{n}\sum_{i=1}^{n} \mathrm{CrsEnt}(h \circ g \circ f(x_i), y_i) + \lambda_1 \mathrm{Clst} + \lambda_2 \mathrm{Sep} \tag{1}$$

ここで P は最適化の対象となるプロトタイプ，w_{conv} は畳み込み層の重み，n はデータのサンプル数，x_i はデータセット内 i 番目のサンプル，y_i は真のラベル，f は CNN の処理を表す関数，g_p はプロトタイプを用いて特徴を表す関数，h は最終的な予測を行うための関数である．また，λ_1 と λ_2 はそれぞれ損失関数に対する重み係数である．式 (1) は 3 つの項から構成され，Clst コストは同じクラスのプロトタイプを近づけ，Sep コストは異なるクラスのプロトタイプを離すために使用する．それぞれの式を式 (2) と式 (3) に示す．

$$\mathrm{Clst} = \frac{1}{n}\sum_{i=1}^{n} \min_{j:p_j \in P_{y_i}} \min_{z \in \mathrm{patches}(f(x_i))} |z - p_j|_2^2 \tag{2}$$

$$\mathrm{Sep} = -\frac{1}{n}\sum_{i=1}^{n} \min_{j:p_j \notin P_{y_i}} \min_{z \in \mathrm{patches}(f(x_i))} |z - p_j|_2^2 \tag{3}$$

この時，z は畳み込み層から出力された特徴マップを表し，P_{y_i} はクラス y のプロトタイプを表し，p_j は最適化の対象となるプロトタイプを表す．式 (2) では各サンプルの特徴マップと真のラベルに対応するプロトタイプとの距離を最小化する．この式によって，正解クラスのプロトタイプは対象のクラスを表現するベクトルになる．式 (3) では各サンプルの特徴マップと真のラベルに対応しないプロトタイプとの距離を最大化する．この式によって，不正解クラスのプロトタイプは異なるベクトルになる．これらから ProtoPNet のプロトタイプは各クラスを表現するように学習する．

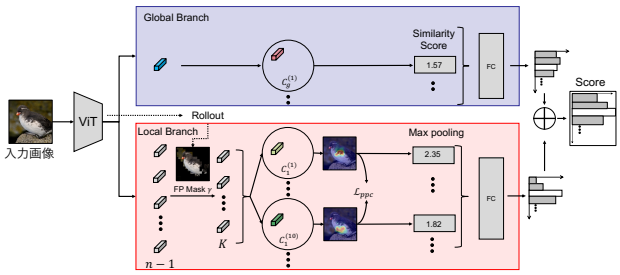

図 1　ProtoPFormer

2.3　ProtoPFormer

ProtoPFormer は ViT を用いたプロトタイプベースのモデルであり，複数のプロトタイプを用いて画像分類を行うことに加えて注目領域を可視化することができる手法である．プロトタイプはあらかじめ表現するクラスを決定している．ProtoPFormer の構造図を図 1 に示す．ProtoPFormer はバックボーンネットワークである ViT, 画像分類や可視化を行うためにプロトタイプを使用している Global Branch と Local Branch の 3 つに分けられる．

ProtoPFormer は最初に ViT に画像を入力することでクラストークンとイメージトークンが出力される．Global Branch では画像全体の情報をもつクラストークンを入力として大域的な特徴とプロトタイプの類似度を算出する．Local Branch では画像の各パッチの情報をもつイメージトークンを入力として各局所的な特徴とプロトタイプの類似度を算出する．その際，バックボーンネットワークの注目領域から作成された FP Mask を用いて前景のイメージトークンのみを対象とする．そして，プロトタイプごと求めた類似度について Max pooling を行う．その後，最大値を求める．最後に，全結合層を適用し，Branch ごとにクラス確率を出力する．最後にそれぞれのクラス確立の平均を求めて出力する．

このモデルは ProtoPNet の基本的損失を継承しながら新たな損失を加えている．加えた損失を式 (4) に示す．

$$L_{\mathrm{PPC}} = \lambda_\mu L^{\mu}_{\mathrm{PPC}} + \lambda_\sigma L^{\sigma}_{\mathrm{PPC}} \tag{4}$$

ここで，L^{μ}_{PPC} は同じクラスのプロトタイプが同じ値にならないように，$L^{\sigma}_{\mathrm{PPC}}$ はプロトタイプが注目する領域を小さくするために使用する．それぞれの式を式 (5) と式 (6) に示す．

$$L^{\mu}_{\mathrm{PPC}} = \frac{1}{(m^c_l)^2} \sum_{i \neq j} \max\left(t_\mu - \|\hat{\mu}^c_i - \hat{\mu}^c_j\|_2, 0\right) \tag{5}$$

ここで，m_l は Local Branch のプロトタイプの数，c はクラス，$\hat{\mu}$ はプロトタイプ，t_μ は閾値である．$\|\hat{\mu}^c_i - \hat{\mu}^c_j\|$ が同じクラスを表現するプロトタイプの類似度である．類似度が閾値 t_μ を超えてしまった場合，プロトタイプが別の領域に注目するように損失を与える．

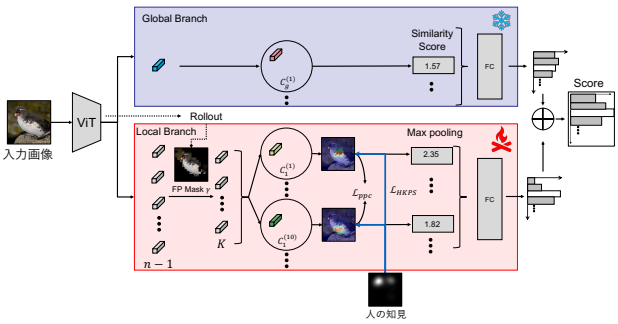

図 2　人の知見の導入による ProtoPFormer の再学習

$$L^{\sigma}_{\mathrm{PPC}} = \mathrm{tr}\left(\max\left(0, \hat{\sum} - t_\sigma\right)\right) \tag{6}$$

ここで $\hat{\sum}$ はプロトタイプの共分散行列の対角成分の平均，t_σ は閾値である．$\hat{\sum}$ の値とプロトタイプが注目する領域は正の相関を持つ

3　提案手法

本節では，従来手法においてモデルが不適切な領域に注目してしまう課題に対処するため，人の知見を活用してモデルを適切な領域に注目させ，精度の向上を図る手法を提案する．また，人の知見を反映させた損失関数の導入方法についても詳しく説明する．

3.1　導入手順

モデルに対して，効果的に人の知見を導入するために局所的な注目をする Local Branch に人の知見を導入する．導入箇所を図 2 に示す．人の知見を用いた損失により，学習するのは Local Branch 内の全結合層とプロトタイプである．提案する損失関数の導入方法を以下に示す．

Step1 **ProtoPFormer の学習**
最初に ProtoPFormer の学習を行い，各プロトタイプが異なる領域に注目するように最適化する．

Step2 **プロトタイプと人の注目領域の比較**
入力画像に対するプロトタイプの注目領域を取得，人が注目する領域との比較を行い，人の知見に近いプロトタイプを選定する．

Step3 **人の知見を用いた再学習**
人の知見に近いプロトタイプ Top N 個を選択，選択したプロトタイプに対して，人の知見を用いた損失を与えながら再度モデルを学習する．

Step4 **繰り返しの終了条件**
クラスごとに精度を確認し，精度が向上したクラスは N の数を 1 増やして再度学習を行う．

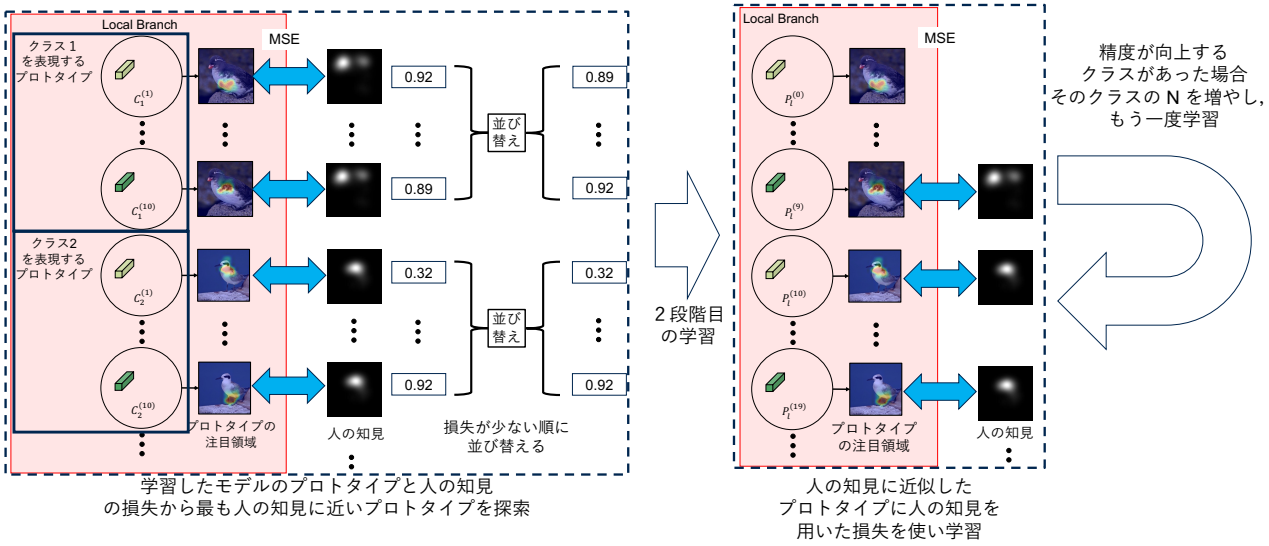

図 3　HKPSLoss の導入方法

Step5 **最後の学習**

それぞれのクラスで最も精度が良かった際のプロトタイプを用いて再度学習を行う．

3.2 Human Knowledge Prototype Superpose Loss

複数のプロトタイプに人の知見を用いた損失を与える Human Knowledge Prototype Superpose Loss（HKPSLoss）を提案する．HKPSLoss の導入を図 3 に示す．学習済みのモデルに対して学習データと人の知見を用いることでプロトタイプと人の知見の誤差を求める．算出された誤差を用いて，人の知見に近いプロトタイプを求める．人の知見に近いプロトタイプ TopN 個のプロトタイプに HKPSLoss を導入し学習する．この時導入する HKPSLoss を式 (7) に示す．

$$L_{HKPS} = \frac{1}{n}\sum_{c}^{Class}\left(\frac{1}{p_c}\sum_{i}^{p_c}(y^i_{pred} - y_{true})^2\right) \quad (7)$$

ここで p_c はクラス c の人の知見を導入するプロトタイプの数である．y_{pred} はプロトタイプの注目領域であり，y_{true} は人の知見である．クラスごとに人の知見を適用するプロトタイプの数が異なるため，$\frac{1}{p_c}\sum_i^{p_c}(y^i_{pred} - y_{true})^2$ によって，各クラスにおける対象プロトタイプと人の知見の損失を求める．また，$Class$ はそれぞれのクラスを示す．これらを用いて各クラスの最も精度が高かった際のプロトタイプを用いて再度学習したモデルを最終的な結果とする．

推論の際は人の知見を用いていないプロトタイプも含めて推論を行う．そのため，人の知見を学習したプロトタイプとモデルが学習によって得たプロトタイプが存在する．

4 評価実験

本章では，提案手法である HKPSLoss の有効性を確認するために，CUB[4] データセットを用いて正解率による定量的評価とプロトタイプの可視化による定性的評価を行う．

4.1 実験条件

CUB データセットには，200 種類の鳥の 9,430/2,357 枚の training/test 画像が含まれており，詳細画像分類に用いられるデータセットである．また，人の知見として，Caltech-UCSD Birds with Gaze and Human Attention（CUB-GHA）データセットを使用する．このデータセットは，CUB データセットの鳥類画像を用いて，人が画像を分類する際の視線データを収集して作成されたものである．特徴抽出器には Data-efficient Image Transformer (DeiT) の tiny モデルを使用する．このモデルは ImageNet-1k[5] を用いて事前学習を行なっている．学習条件として最適化手法は AdamW optimizer[6] と cosine LR schduler，エポック数は 200，バッチサイズは 128，ViT から出力される Image Token は 196 個，FP Mask では 81 個の Image Token を抽出する．また，使用するプロトタイプの数は各クラス 10 個使用する．これらの条件で学習した後，人の知見を活用した損失を追加導入するためにエポック数を 220 に変更し，追加学習を行う．

4.2 人の知見を導入するプロトタイプの数による精度変化

人の知見を導入するプロトタイプを増やしながら学習したモデルの正解率の変化を図 5 に示す．人の知見を導入した直後が最も精度が向上していることがわかる．そのため，人の知見に近い注目をするプロトタイプが一つある場合，適切に分類することができる画像

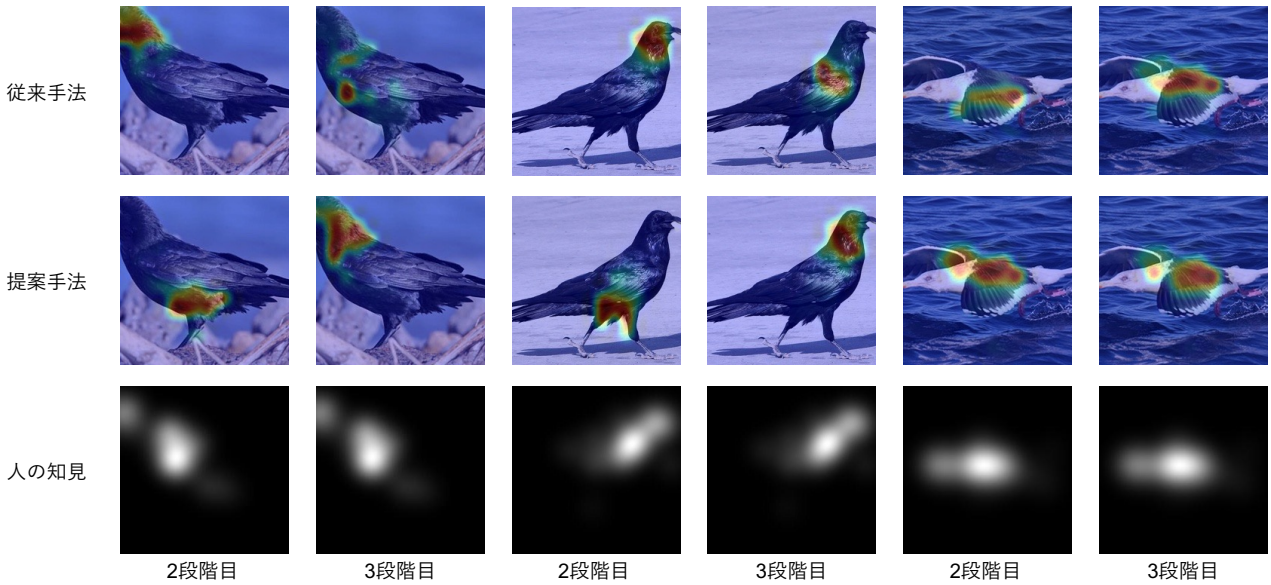

図 4　HKPSLoss による人の知見を導入する段階を変えた際の可視化結果

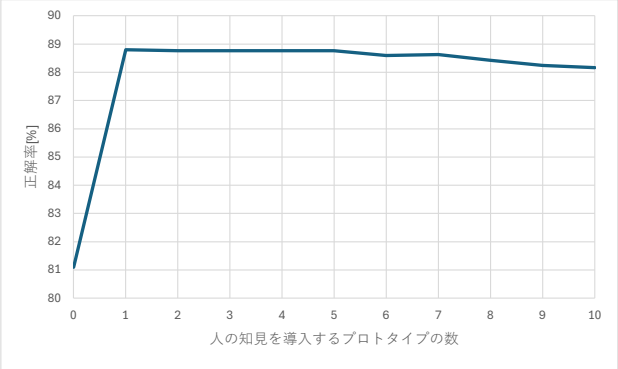

図 5　人の知見の導入による精度変化

が複数枚あることがわかる．また，より多くのプロトタイプに人の知見を導入するほど正解率が低下している．しかし，クラスごとの精度では上昇しているクラスが存在していることから，適切に人の知見を導入することでより効果的に学習ができると考えれられる．

4.3　HKPSLoss の定性的評価

人の知見による損失の導入による注目領域の変化を確認する．可視化には，入力画像から抽出された Image トークンとプロトタイプとの類似度を計算し，その結果を入力画像に重ね合わせて行う．可視化する画像は，従来手法では分類に失敗したが，HKPSLoss によって分類が成功した画像を対象とする．また，可視化に使用するプロトタイプは最初に人の知見を導入したプロトタイプと次に人の知見を導入したプロトタイプを用いる．HKPSLoss を用いた学習したモデルを可視化した結果を図 4 に示す．

表 1　従来手法と人の知見を加えた際の精度比較

	Accuracy	F1-score
ProtoPFormer	81.10	80.82
提案手法	89.69	88.81

4.4　損失関数による正解率の比較

提案手法を用いて学習を行った際の CUB-200-2011 に対しての正解率と F1-score を表 1 に示す．提案手法では従来手法と比べ Accuracy は 8.59pt, F1-score は 7.99pt の向上を確認できた．プロトタイプに対して人の知見を用いた損失により精度が向上したことから，人が判別する際に注目した領域はモデルが画像分類する際にも有効であると考えられる．これらの結果から，人の知見を導入することで，認識モデルは人が注目すべきと考える領域に注目することで精度向上が可能である．

左から 1 枚目の画像は 2 段階目の学習で人の知見を導入したプロトタイプであり，元々最も人の知見に近いプロトタイプであった．しかし，3 段階目の学習で人の知見を導入したプロトタイプの方が，人の知見により近い注目をしていることがわかる．

左から 3 枚目の画像は別の画像に対するプロトタイプの画像である．こちらも 2 段階目と 3 段階目の学習時の画像を比較すると 3 段階目の画像の方が人の知見に近い注目をしていることがわかる．

これらの注目は 3 段階目の学習で HKPSLoss を適用したプロトタイプによって，最初に HKPSLoss を適用していたプロトタイプは，より精度向上につながる重要な領域へ注目を移したと考えられる．これは，3 段階目の学習で HKPSLoss を適用したプロトタイプによっ

て，2 段階目に HKPSLoss を適用していたプロトタイプは，人の知見に従った領域に注目するよりも別の領域に注目をしたほうが精度が向上すると学習したからだと考えられる．

左から 5 枚目の画像では 2 段階目の学習によって先に人の知見を学習したプロトタイプの方が 3 段階目の学習でのプロトタイプの注目領域よりも人の知見に近い注目をしている．また，2 つの注目領域は近い領域に注目している．

図 4 の左側 2 種類の画像の場合，このクラスでは人が分類する際には鳥の首付近に注目するが，モデルは鳥の足の付け根部分が最も重要であると判断したと考えられる．また，最も右側の画像では人が注目した領域が注目するべき領域と判断し，2 つのプロトタイプは似た注目をしている．これらから，HKPSLoss を用いることで，人が注目する領域が重要であればモデルはその領域を注目するが，人の注目領域以上に重要な領域があればモデルは別の注目をする．そのため，人が注目する領域とモデルが重要だと考え注目する領域を同時に表現することが可能になったと考えられる．

5 結論

本論文では，ViT にプロトタイプを用いたモデルに対して，人の知見を活用した損失関数である HKPSLoss を提案した．実験結果から，これらの損失関数は分類精度の向上に有効であることが示された．さらに，注目領域が人の知見に近づくことで，可視化結果が直感的に理解しやすくなり，モデルの解釈容易性の向上にも寄与することが確認された．また，複数のプロトタイプに人の知見を適用することで，人が分類時に用いる判断根拠をモデルが保持しつつ，別の重要な領域も表現できるようになった．これにより，モデルは解釈容易性を向上させるだけでなく，新たな知見を人に提供する可能性も示唆された．一方で，人の知見を導入するプロトタイプの組み合わせや適用順序については最適解が明確ではなく，評価方法によっては異なる知見が得られる可能性やさらなる精度向上が期待される．これらの課題を踏まえ，今後は，人の知見を導入するための評価式の設計や適用順序の最適化などを進めることで，より高い分類精度と注目領域の言語化などを用いることで解釈容易性の明確な向上を目指す．

参考文献

[1] K. He, X. Zhang, S. Ren, and J. Sun, ”Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[2] A. Vaswani et al., ”Attention is all you need,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.

[3] A. Dosovitskiy et al., ”An image is worth 16x16 words: Transformers for image recognition at scale,” in *Int. Conf. Learn. Represent. (ICLR)*, 2021.

[4] P. Welinder et al., ”Caltech-ucsd birds 200,” *Caltech, Tech. Rep. CNS-TR-201*, 2010.

[5] J. Deng et al., ”Imagenet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 248–255.

[6] I. Loshchilov and F. Hutter, ”Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.

[7] M. Xue et al., ”Protopformer: Concentrating on prototypical parts in vision transformers for interpretable image recognition,” *arXiv preprint arXiv:2203.XXXXX*, 2022.

[8] C. Chen et al., ”This looks like that: Deep learning for interpretable image recognition,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019.