Vision-language基盤モデルの画像破損に対するテスト時適応

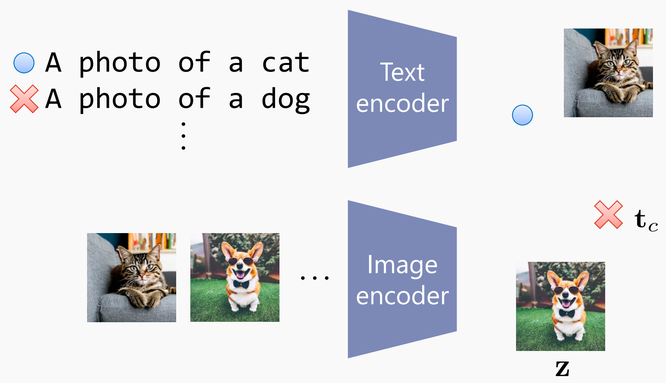
arXiv GitHub

足立 一樹*†，山口 真弥*，濱上 知樹†
*NTT株式会社 †横浜国立大学

Vision-language基盤モデル

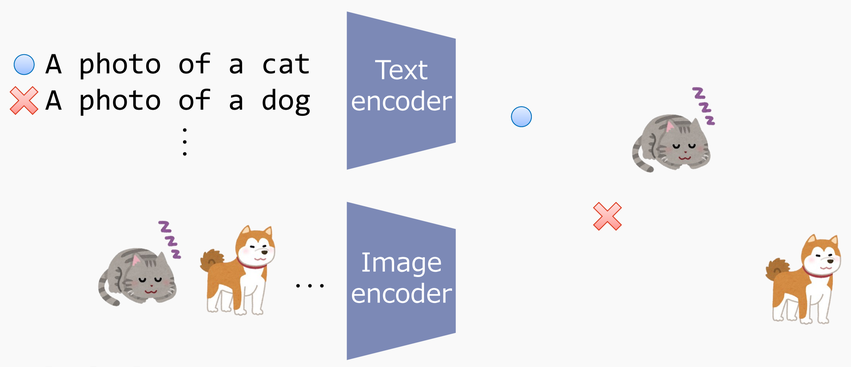
- **Vision-language基盤モデル (VLM)**
  - ➢ 画像とテキストのペアデータを学習した基盤モデル
    - › 代表例: CLIP
  - ➢ 画像とテキストを共通の埋め込み空間に写像することで統一的に扱うことが可能に
  - ➢ 画像エンコーダとテキストエンコーダを持つ
- **ゼロショット分類**
  - ➢ 学習済みVLMを用いて、追加学習なしで画像分類モデルを得る手法
  - ➢ クラスを表すテキスト埋め込みとの類似度で分類

$$\hat{p}_c = \mathrm{softmax}\left([\cos_s(\mathbf{z}, \mathbf{t}_1), \ldots, \cos_s(\mathbf{z}, \mathbf{t}_C)]\,/\tau\right)_c$$



テスト時適応 (TTA)

- **分布シフトによりゼロショット分類の精度が低下**
  - ➢ プロンプトが合わなくなるため



- **テスト時適応 (Test-time adaptation; TTA)**
  - ➢ テストデータ（ラベルなし）のみを使い推論しながら分布シフトに適応
  - ➢ 既存手法: 埋め込みベクトルや予測確率を事後的に補正
    - › Test-time prompt tuning (Shu et al., NeurIPS2022)
    - › Training-free dynamic adapter (Karmanov et al., CVPR2024)

課題

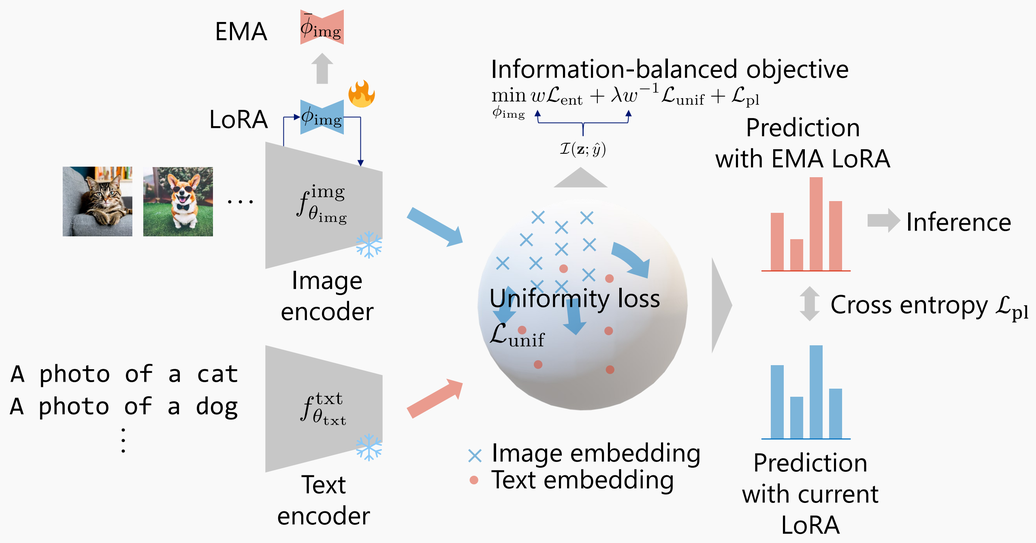
- **VLMは画像破損に対して性能が大きく低下**
  - ➢ 既存手法はドメインシフトには有効だが、画像破損に対しては効果が低い

- **ドメインシフトと画像破損の違いを分析**
  - ➢ 画像埋め込みのUniformity (一様分布度合い) が低下
  - ➢ 画像埋め込み同士の判別性が低下している
    → 画像エンコーダを更新するべき

$$\text{Uniformity loss} := \mathbb{E}_{\mathbf{z}_1, \mathbf{z}_2}\left[\exp\left(-\|\mathbf{z}_1 - \mathbf{z}_2\|_2^2\right)\right]$$

Metric	ImageNet	ImageNet-A	ImageNet-R	Domain shift Mean	Defocus blur	Glass blur	Motion blur	Zoom blur	Contrast	Elastic transform	Jpeg compression	Pixelate	Gaussian noise	Impulse noise	Shot noise	Brightness	Fog	Frost	Snow	Corruption Mean
Accuracy (Normal prompt, ↑)	66.97	32.91	74.36	58.08	28.18	11.81	19.13	17.65	17.90	13.37	36.77	36.92	6.02	6.12	7.88	54.75	34.39	27.32	27.77	23.06
Accuracy (Ensemble, ↑)	67.59	33.15	76.51	59.08	29.09	12.56	20.56	19.12	18.55	14.34	37.71	37.89	6.36	6.54	8.27	55.78	35.99	28.37	28.34	23.97
Accuracy (Corruption prompt, ↑)	-	-	-	-	26.53	12.15	18.39	17.99	17.65	13.96	36.29	36.35	6.02	6.10	7.80	53.93	33.43	27.34	26.55	22.70
Entropy (↓)	0.748	1.220	0.595	0.854	2.011	2.703	2.317	2.245	3.247	2.297	1.632	1.615	3.773	3.714	3.671	1.061	1.681	1.905	2.048	2.395
Uniformity loss (↓)	0.513	0.538	0.500	0.517	0.682	0.735	0.722	0.715	0.744	0.706	0.630	0.641	0.855	0.853	0.839	0.601	0.665	0.655	0.686	0.715
Modality gap (EMD, ↓)	1.291	1.333	1.348	1.324	1.298	1.326	1.325	1.327	1.337	1.341	1.293	1.296	1.299	1.297	1.296	1.293	1.307	1.315	1.308	1.311

提案手法



- **基本アイデア**
  - ➢ 画像埋め込みのUniformity改善＋エントロピー最小化
  - ➢ 画像エンコーダのパラメータを更新

$$\min_{\theta_{\text{img}}} \mathcal{L}_{\text{ent}} + \lambda \mathcal{L}_{\text{unif}}$$

$$\mathcal{L}_{\text{ent}} := \frac{1}{B}\sum_{i=1}^{B}\sum_{c=1}^{C} -\hat{p}_c \log \hat{p}_c, \quad \mathcal{L}_{\text{unif}} := \log \frac{1}{B^2}\sum_{i=1}^{B}\sum_{j=1}^{B} \exp\left(\|\mathbf{z}_i - \mathbf{z}_j\|_2^2\right)$$

- **Uniformityとエントロピーのバランス調整**
  - ➢ 相互情報量をもとにバランス調整
    - › 小さい：分類が上手く行っていない → Uniformityを優先
    - › 大きい：エントロピーを優先

$$\min_{\theta_{\text{img}}} w\mathcal{L}_{\text{ent}} + \lambda w^{-1}\mathcal{L}_{\text{unif}}$$

$$w = \exp(\mathcal{I}(\mathbf{z}; \hat{y}) - \mathcal{I}_0)$$

$$\mathcal{I}(\mathbf{z}; \hat{y}) = \mathcal{H}(\mathbf{z}) - \mathcal{H}(\mathbf{z}|\hat{y})$$

$$= \sum_{c=1}^{C} -\bar{p}_c \log \bar{p}_c - \frac{1}{B}\sum_{i=1}^{B}\sum_{c=1}^{C} -\hat{p}_{i,c} \log \hat{p}_{i,c}$$

実験

- **画像破損下でゼロショット分類精度を比較**
  - ➢ 提案手法により精度が向上
  - ➢ 既存手法は逆に精度が低下するケースもあり

Method	Defocus blur	Glass blur	Motion blur	Zoom blur	Contrast	Elastic transform	Jpeg compression	Pixelate	Gaussian noise	Impulse noise	Shot noise	Brightness	Fog	Frost	Snow	Mean
No-adapt	28.31	11.89	19.16	17.61	17.87	13.22	36.79	37.01	6.08	6.17	7.85	54.89	34.55	27.35	27.67	23.09
Linear probing (1)	11.53	6.15	8.79	9.02	5.87	8.17	15.57	16.36	2.79	2.76	3.41	27.05	16.43	9.88	11.87	10.38
Linear probing (5)	19.55	10.53	15.16	15.25	10.04	14.47	25.68	27.02	4.76	4.95	5.62	43.28	26.10	17.31	19.48	17.28
Linear probing (10)	23.84	12.72	17.97	18.57	12.45	17.69	30.50	32.32	5.78	6.04	6.83	49.90	31.15	21.49	23.63	20.73
Tip-adapter (Zhang et al., 2022) (1)	19.00	9.09	13.92	14.04	9.53	12.60	26.98	27.19	4.06	4.02	5.15	44.92	25.99	18.10	19.28	16.92
Tip-adapter (5)	23.43	12.27	17.61	17.76	11.56	16.82	30.87	32.09	4.88	4.86	6.03	49.88	30.59	21.43	22.82	20.19
Tip-adapter (10)	26.11	13.99	19.85	20.23	12.99	19.26	33.03	34.60	5.52	5.82	6.82	52.40	33.15	24.01	24.83	22.17
TPT (Shu et al., 2022)	29.66	12.87	21.11	20.54	20.11	15.21	39.27	41.14	6.48	6.74	8.50	57.35	37.05	29.81	30.23	25.07
ZERO (Farina et al., 2024)	26.85	8.86	18.11	19.89	16.46	12.38	35.09	37.44	3.69	5.33	4.43	53.35	33.50	26.56	27.42	21.96
MTA (Zanella & Ayed, 2024)	27.79	11.29	19.25	18.88	21.18	13.92	37.23	38.95	2.41	2.87	2.96	53.56	34.32	28.02	28.66	22.75
TDA (Karmanov et al., 2024)	30.13	14.59	22.10	**21.09**	19.59	**17.15**	38.58	39.53	7.23	7.45	9.34	56.99	38.09	30.24	31.02	25.54
UnInfo (ours)	**31.51**	**16.76**	**23.47**	20.40	**22.81**	16.59	**42.03**	**42.38**	**7.56**	**10.60**	**11.36**	**57.75**	**39.16**	**31.65**	**32.40**	**27.10**

- **TTA前後で埋め込みベクトルを可視化**
  - ➢ 提案手法によりUniformityが改善
    → 画像埋め込みとテキスト埋め込みのギャップが解消