자기 도전적 메커니즘과 특징 분해를 통한 교차 도메인 일반화 성능 개선

송승헌, 이재구*
국민대학교

*jaekoo@kookmin.ac.kr

Enhanced Cross-Domain Generalization through Self-Challenging Mechanism and Feature Decomposition

Seungheon Song, Jaekoo Lee*
College of Computer Science, Kookmin University

요 약

도메인 일반화는 학습 과정에서 보지 못한 도메인의 데이터에 적응할 수 있도록 하는 과업이며, 딥러닝이 실제 세계에서 잘 작동하기 위해 풀어야 할 필수적인 문제이다. 우리는 도메인 일반화를 달성하기 위해 도메인 분기 및 클래스 분기의 그레디언트를 활용하여 학습하는 SCFA(Self-Challenging and Feature Augmentation)를 제안한다. 또한, 특징 분포를 확장하고 도메인 불변 특징을 학습하기 위해 자기 도전 방법과 특징 분해를 통한 증강을 사용한다. 결과적으로 SCFA 는 기준선에 비해 PACS 데이터 집합에서 2.47%, OfficeHome 데이터 집합에서 1.79% 향상된 정확도를 보여주었다.

Ⅰ. 서 론

실제 세계에서 인간의 시각 체계는 객체에 대한 특징을 파악하고 서로 다른 객체를 분류할 수 있으며, 보지 못한 도메인 (Domain)에 대해서도 객체를 분류할 수 있는 능력을 가지고 있다. 하지만 딥러닝 (Deep Learning) 모델은 학습 과정에서 보았던 도메인의 데이터에서는 분류를 잘하더라도, 학습 과정에서 보지 못한 도메인의 데이터에 대해서는 분류 성능이 크게 떨어지게 된다[1].

성능이 떨어지는 대표적인 원인은 학습 환경과 추론 환경 간의 차이이다. 일반적인 딥러닝 모델들의 경우 학습 도메인과 추론 도메인이 유사한, IID (Independent and Identically Distributed) 환경에 있다고 가정하고 학습을 하게 된다. 하지만 실제 세계의 데이터는 학습 도메인과 추론 도메인이 다른, OOD (Out Of Distributed) 환경에 존재하는 경우가 많기 때문에, 학습 과정에서 보지 못한 도메인에 대해 취약해지게 된다. 이를 해결하기 위해 최근 딥러닝 연구는 원본 도메인에서 학습시킨 후, 추론 과정에는 학습 과정에서 보지 못한 OOD 환경의 데이터인 목표 도메인에 대해서 좋은 성능을 보이도록 하는 도메인 일반화 (Domain Generalization) 방법들이 많이 연구되고 있다.

도메인 일반화를 위한 최신 연구 중에는 데이터 증강 (Data Augmentation)을 활용하여 원본 도메인의 분포를 확장하는 방식[2] 혹은 적대적 학습방법을 사용하는 방식[3]으로 도메인 불변 특징 (Domain Invariant Feature)을 학습하는 방법이 사용되고 있다. 우리의 연구는 그레디언트 (Gradient)를 이용하여 도메인 불변 특징을 학습하는 방법인 RSC[4], DecAug[5] 등에 영감을 받아 SCFA(Self-Challenging and Feature Augmentation)를 제안한다.

SCFA 에서는 도메인 분기 (Domain Branch)를 이용해서 도메인을 예측하고, 클래스 분기 (Class branch)와 혼합 분기 (Fusion Branch)를 이용해서 클래스를 예측한다. 또한, 자기 도전적인 방법 (Self-Challenging Mechanism)[4]을 통해 클래스 분기의 특징 분포를 확장하고, 도메인 분기의 특징에 섭동 (Perturbation)을 주어 도메인 불변 특징을 추출하는 방법을 제안한다. 이러한 방법을 통해 PACS[6] 데이터 집합에서 2.47%, OfficeHome[7] 데이터 집합에서 1.79%의 정확도 향상을 보였다.

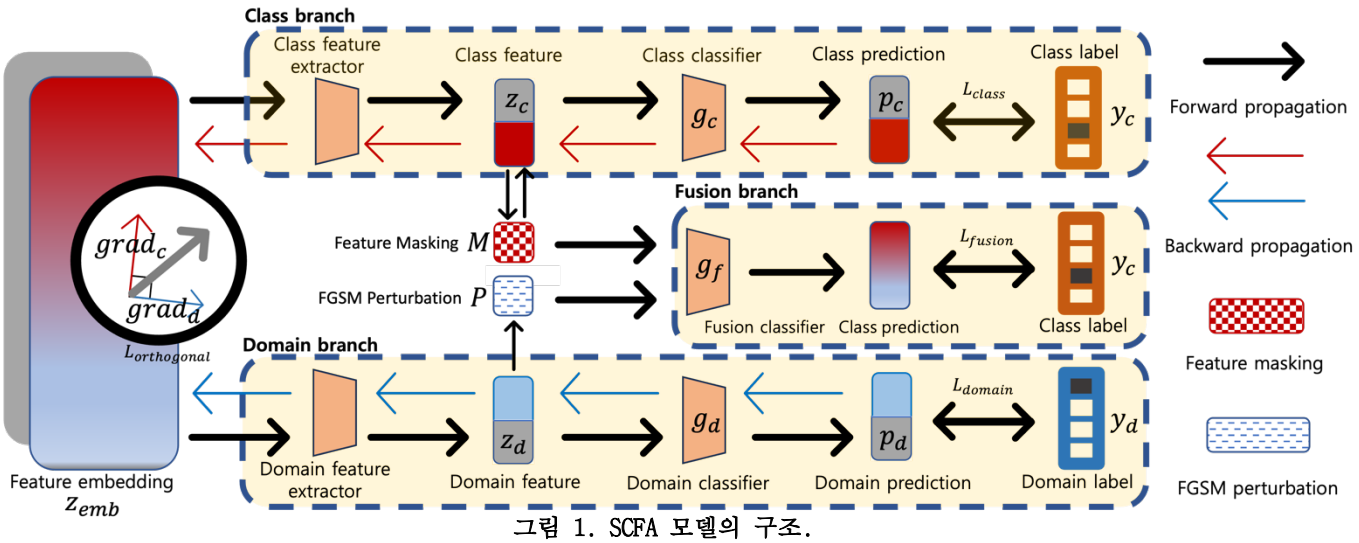

그림 1. SCFA 모델의 구조.

표 1. PACS[6] 데이터 집합과 OfficeHome[7] 데이터 집합에서 도메인 일반화 성능 측정 결과(정확도)

Dataset	PACS[6]					OfficeHome[7]				
Domain	Artpaint	Cartoon	Photo	Sketch	Avg	Art	Clipart	Product	Real	Avg
RSC[4]	84.66	82.23	92.30	77.72	84.23	59.79	**52.89**	72.57	73.80	64.76
Ours	**88.59**	**83.20**	**97.00**	**78.02**	**86.70**	**61.80**	51.23	**75.62**	**77.54**	**66.55**

Ⅱ. 본 론

클래스와 도메인 간의 상관관계가 존재하는 경우 클래스 특징과 도메인 특징은 상관관계를 가지게 된다[5]. 이러한 상관관계는 도메인 일반화 과업에서 도메인 불변특징을 학습하는데 장애가 될 수 있다. SCFA 에서는 [그림 1]과 같이 클래스 특징 (Class Feature)과 도메인 특징 (Domain Feature)의 상관관계를 끊기 위해 클래스 분기와 도메인 분기를 이용한다. 또한, 그레디언트값을 학습에 이용하기 위해 순전파 (Forward Propagation)와 역전파 (Backward Propagation)를 한차례 진행한다. 순전파 과정에서는 특징 추출기로부터 나온 특징 임베딩 (Feature Embedding) z_{emb}이 클래스 분기와 도메인 분기로 입력되어 각각 클래스 레이블 (Class Label) y_c과 도메인 레이블 (Domain Label) y_d을 예측하도록 한다. 이렇게 예측한 값을 통해 특징 임베딩 z_{emb}에 클래스 그레디언트 $grad_c$와 도메인 그레디언트 $grad_d$가 쌓이게 된다. 이 두 그레디언트를 이용하여 두 특징이 서로 수직하게 만들도록 유도하고 클래스 특징이 도메인에 무관하게 클래스를 예측할 수 있도록 하기 위해 [식 1]과 같은 수직 손실 $L_{orthogonal}$을 사용한다.

$$L_{orthogonal} = \left(\frac{grad_c}{|grad_c|} \cdot \frac{grad_d}{|grad_d|}\right)^2 \quad (1)$$

[그림 1]의 클래스 분기는 입력의 클래스를 맞추기 위한 특징을 학습한다. 이때, 정답 레이블을 예측하는데 영향을 많이 준 특징을 마스킹 (Masking) 한 후 마스킹 된 특징을 이용하여 정답 레이블을 예측하도록 하는 자기 도전적인 방법을 사용한다. 앞서 진행한 역전파 과정에서 클래스 특징에 쌓인 그레디언트중 상위 33%의 그레디언트에 마스킹을 적용한다. 이러한 자기 도전적인 방법은 입력의 풍부한 특징을 학습하도록 유도하여 클래스 특징의 분포를 확장한다. 특징을 학습하기 위한 클래스 예측 손실은 [식 2]와 같다.

$$L_{class} = -\sum_i y_i^c \log p(g_c(M(z_c))) \quad (2)$$

[그림 1]의 도메인 분기는 도메인 레이블을 맞추기 위한 특징을 학습한다. 도메인 분기는 클래스 특징과 도메인 특징이 수직이 되도록 하기 위해 사용된다. 도메인 레이블을 예측하도록 하는 도메인 손실은 [식 3]과 같다.

$$L_{domain} = -\sum_i y_i^d \log p(z_{emb})) \quad (3)$$

또한, 다양한 OOD 환경에서 클래스 특징과 도메인 특징의 상관관계를 제거하기 위해 도메인 특징에 데이터 증강을 적용한다. 이렇게 증강된 특징은 혼합 분기에 입력으로 사용하게 된다. 데이터 증강을 위한 기법으로는 FGSM 섭동 (Fast Gradient Sign Method Perturbation)[8]을 사용한다.

[그림 1]의 혼합 분기는 자기 도전적 방법에 의해 마스킹 된 클래스 특징과 증강된 도메인 특징을 결합하여 사용한다. 혼합 손실은 결합된 특징을 이용하여 [식 4]와 같이 클래스를 예측하고, 클래스와 도메인의 다양한 상관관계를 끊고 다양한 특징을 학습할 수 있도록 한다.

$$L_{fusion} = -\sum_i y_i^c \log p(g_f([M(z_c)||P(z_d)])) \quad (4)$$

최종적으로 모델의 학습을 위해서 [식 5]와 같이 위의 4 개의 손실을 모두 합한 최종 손실 L_{total}를 이용하여 모델을 학습하게 된다.

$$L_{total} = L_{domain} + L_{orthogonal} + L_{class} + L_{fusion} \quad (5)$$

Ⅲ. 실험 및 결과

SCFA 의 도메인 일반화 성능을 평가하기 위해 PACS[6] 데이터 집합과 OfficeHome[7] 데이터 집합에서 LOO (Leave-One-Out)방법을 사용하여 기존 모델인 RSC 와 정확도를 비교 측정하였다. [표 1]에서는 테스트 집합을 열 이름으로 표기하였다. SCFA 는 [표 1]에서와 같이 PACS[6] 데이터 집합에서 평균 86.70%의 정확도를 얻어 기준선인 RSC[4]에 비해 2.47%, OfficeHome[7] 데이터 집합에서 평균 66.55%의 정확도를 얻어 1.79%의 정확도 향상을 얻을 수 있었다.

Ⅳ. 결 론

본 논문에서는 OOD 환경에 강건한 모델인 SCFA 를 제안하였다. SCFA 는 그레디언트를 이용하는 자기 도전적 방법으로 특징의 분포를 확장하여 도메인 변화에 강건한 특징을 만들었다. 또한, 클래스 특징과 도메인 특징을 수직하게 하여 클래스 특징과 도메인 특징을 구분하였으며, 특징 차원의 데이터 증강을 사용하여 도메인 불변 특징들을 학습했다. 이러한 방법들을 통해 기준선에 비해 OOD 환경에 강건한 모델을 만들 수 있었다. 결과적으로 기준선에 비해 두 데이터 집합 모두에서 정확도가 향상되었다.

ACKNOWLEDGMENT

이 논문은 2022 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.RS-2022-00167194, 미션 크리티컬 시스템을 위한 신뢰 가능한 인공지능)

참 고 문 헌

[1] Li, D., et al. "Deeper, broader and artier domain generalization." Proceedings of the IEEE international conference on computer vision. 2017.

[2] Volpi, Riccardo, et al. "Generalizing to unseen domains via adversarial data augmentation." Advances in neural information processing systems 31 2018.

[3] Li, Haoliang, et al. "Domain generalization with adversarial feature learning." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[4] Huang, Zeyi, et al. "Self-challenging improves cross-domain generalization." Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. Springer International Publishing, 2020.

[5] Bai, Haoyue, et al. "Decaug: Out-of-distribution generalization via decomposed feature representation and semantic augmentation." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35. No. 8. 2021.

[6] Zhou, Kaiyang, et al. "Deep domain-adversarial image generation for domain generalisation." Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 07. 2020.

[7] Rahman, Mohammad Mahfujur, et al. "Multi-component image translation for deep domain generalization." 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2019.