Keypoint Detection과 Annoy Tree를 사용한 2D Hand Pose Estimation

이희재[O], 강민혜[*]
[O]금오공과대학교 컴퓨터공학과,
[*]금오공과대학교 컴퓨터공학과
e-mail: lhj5682@kumoh.ac.kr[O], Gangminhye77@kumoh.ac.kr[*]

Fast Hand Pose Estimation with Keypoint Detection and Annoy Tree

Hui-Jae Lee[O], Min-Hye Kang[*]
[O]Dept. of Computer Engineering, Kumoh National Institute of Technology,
[*]Dept. of Computer Engineering, Kumoh National Institute of Technology

● 요 약 ●

최근 손동작 인식에 대한 연구들이 활발하다. 하지만 대부분 Depth 정보를 포함한3D 정보를 필요로 한다. 이는 기존 연구들이 Depth 카메라 없이는 동작하지 않는다는 한계점이 있다는 것을 의미한다. 본 프로젝트는 Depth 카메라를 사용하지 않고 2D 이미지에서 Hand Keypoint Detection을 통해 손동작 인식을 하는 방법론을 제안한다. 학습 데이터 셋으로 Facebook에서 제공하는 InterHand2.6M 데이터셋[1]을 사용한다. 제안 방법은 크게 두 단계로 진행된다. 첫째로, Object Detection으로 Hand Detection을 수행한다. 데이터셋이 어두운 배경에서 촬영되어 실 사용 환경에서 Detection 성능이 나오지 않는 점을 해결하기 위한 이미지 합성 Augmentation 기법을 제안한다. 둘째로, Keypoint Detection으로 21개의 Hand Keypoint들을 얻는다. 실험을 통해 유의미한 벡터들을 생성한 뒤 Annoy (Approximate nearest neighbors Oh Yeah) Tree를 생성한다. 생성된 Annoy Tree들로 후처리 작업을 거친 뒤 최종 Pose Estimation을 완료한다. Annoy Tree를 사용한 Pose Estimation에서는 NN(Neural Network)을 사용한 것보다 빠르며 동등한 성능을 냈다.

키워드: Image composition, Image Augmentation, Keypoint Detection, Annoy Tree

I. Introduction

최근 손동작 인식에 대한 연구들이 활발하다. 하지만 대부분 Depth 정보를 포함한 3D 정보를 필요로 한다. 이에 Depth 카메라가 없는 사람들도 손동작을 통한 유용한 소프트웨어를 사용할 수 있게 하고자 본 연구를 시작하게 되었다.

Detection 부분에서는 이미지 합성 Augmentation을 통한 실 사용 환경에서의 Hand Detection 성능 향상을 목표로 하였다.

Annoy Tree를 사용한 Pose Estimation 부분에서는 단순 Neural Network 보다 빠르며 성능도 좋게하는 것을 목표로 하였다.

II. The Proposed Scheme

1. Image Augmentation

기본 데이터셋으로만 학습시켰을 때 실제 환경에서 inference 해보면 얼굴도 손으로 인식하고 검은 배경이 아닌 환경에서는 detection이 잘 되지 않는 문제가 있었다. 이를 학습 데이터셋의 배경이 주로 검은색이기 때문이라고 보고 이미지 합성 data augmentation으로 배경 학습을 하였다. [그림 1]이 이미지 합성의 결과이다. 순서대로 원본 데이터, 원본데이터에 WIDER FACE[3] 데이터를 합성한 데이터, 원본데이터에 얼굴과 단색 배경을 합성한 데이터이다.

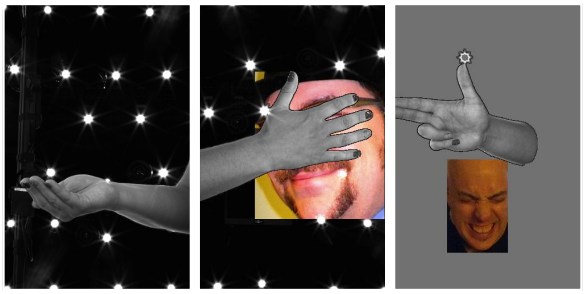

Fig. 1. 이미지 합성 결과물

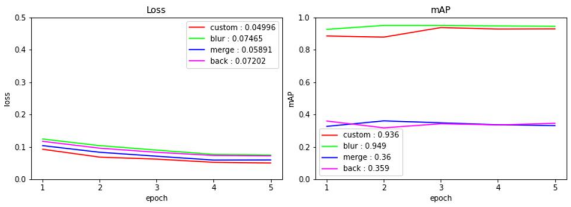

Fig. 2. 학습결과

[그림 2]는 [그림 1]로 구축한 학습세트로 학습한 결과이다. custom, blur는 원본 데이터와 blur augmen-tation을 추가하여 학습한 것이고, merge는 얼굴만 합성한 데이터, back은 얼굴과 배경을 합성한 데이터로 학습한 것이다. loss는 4개의 학습 모두 0.05 ~ 0.07 수준으로 큰 차이가 없었으나 mAP에서 큰 차이가 났다. Custom과 blur는 0.93까지 올랐으나 이미지를 합성한 merge와 back에서 0.36까지 밖에 나오지 않으며 오히려 더 안 좋은 결과가 나왔다.

2. Pose Estimation with Annoy Tree

Keypoint 사이의 거리들을 조합하여 만든 벡터[그림 3]로 Annoy Tree를 생성하여 분류한 것과 Keypoint 좌표 자체를 Neural Network를 사용하여 분류한 것을 비교해 보았다[표 1]. Pose는 총 5가지로 자체 데이터 셋을 구축하여 총 150장으로 테스트 하였다. Accuracy는 동등하였고 처리시간은 Annoy Tree가 조금 앞섰다.

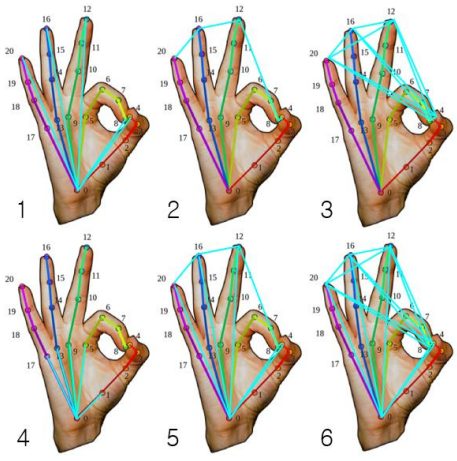

Fig. 3. 벡터 조합

Table 1. Annoy Tree와 NN 성능비교

	Annoy Tree	NN
Accuracy(%)	91.33	91.33
Time(sec)	0.000117788	0.000189611

III. Conclusions

이미지 합성을 통해 실 사용 환경에서의 성능 향상을 목표로 하였으나 결과가 안 좋았다, 프레임 워크 특성상 모델을 fine tuning 하기 힘들고 학습과정을 상세히 확인하기 어려워 원인을 찾지 못했지만 향후에 최신 Detector를 직접 사용하여 원인을 분석하고 성능을 끌어올릴 예정이다.

Pose Estimation은 좋은 결과가 나왔지만, Detection 부분과 Keypoint Detection 부분에서 소요되는 시간을 고려하면 Annoy Tree를 사용하여 얻는 이득이 미미하다고 볼 수 있다. 하지만 Keypoint Detection이 잘못된 경우에도 딥러닝을 사용하지 않고도 후처리를 통해 성능을 올릴 수 있다는 점과 classification을 추가 학습하지 않아도 된다는 점에 의의가 있다. 또한, 이 실험에서는 다섯가지 동작만을 구분하였지만 Keypoint 벡터들을 조합하여 가능한 모든 손동작을 구분할 수 있도록 발전시킬 수 있다.

REFERENCES

[1] Gyeongsik Moon, ShoouI Yu, He Wen, Takaaki Shiratori, Kyoung Mu Lee, “InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image, In: ECCV (2020)

[2] “MMDetection”, github, https://github.com/open-mmlab/mmdetection(2020)

[3] Shou Yang, Ping Luo, Chen Change Loy, Xiaoou Tang, “WIDER FACE: A Face Detection Benchmark”, CVPR(2016)

[4] “MMPose”, github, https://github.com/open-mmlab/mmpose(2020)

[5] “ANNOY”, github, https://github.com/spotify/annoy (2019)

[6] Arya, S., Mount, D. M., Netanyahu, N.S., Silverman, R., Wu, A., “An optimal algorithm for approximate nearest neighbor searching”, In: ACM(2009)