|  |
| --- |
| **FaceNet의 Fine-Tuning을 통한**  **얼굴 검출 웹 서비스**  김현곤  서강대학교 컴퓨터공학과  [hgk93@sogang.ac.kr](mailto:hgk93@sogang.ac.kr) |
|  |
|  |
| **요 약**  최근 많은 얼굴 인식 기술이 개발, 출시되며 얼굴 인식에 대한 관심이 올라가고 있으며, 합성곱 신경망을 이용한 많은 방법들의 연구가 진행 되고 있다. 얼굴 검출에서 대중적으로 쓰이는 FaceNet은 학습 데이터가 외국인이기 때문에 한국인을 대상으로 하였을 때 좋은 결과를 얻기 힘들 수 있다. 따라서 좋은 결과를 얻기 위한 방법으로 FaceNet의 분류기만을 Fine-tuning 하였을 때와 미리 학습된 특징 추출기와 분류기를 같이 Fine-tuning 하였을 때의 성능을 실험을 통해 비교 분석하였다. 한국 연예인 1,000명을 대상으로 분류기만을 Fine-tuning 하였을 때는 90.3%의 정답률을 보였지만, 특징 추출기와 분류기를 같이 Fine-tuning 하였을 때는 91.3%의 정답률을 보였다. 연예인 수가 적은 실험에서는 훨씬 높은 정답률을 보였다. 앞에서 학습한 특징 추출기와 분류기를 이용하여 Django 기반의 얼굴 검출 웹 서비스를 제작하였다. | | |

**1. 서 론**

최근 미디어 자원이 기하급수 적으로 증가함에 따라 미디어를 이용하는 합성곱 신경망(Convolutional Neural Network) 을 이용한 연구들이 활발히 진행되고 있으며, 이 중 하나인 얼굴 검출 기술도 함께 연구되고 있다. 얼굴 검출 문제는 이미지에 있는 사람의 얼굴이 누구인지 알아내는 문제로, 합성곱 신경망을 사용하여 높은 성능을 끌어낼 수 있지만 학습에 사용되는 벤치마크 데이터가 CASIA-Webface[1], MS-celeb-1M[2]와 같이 외국인인 경우가 대다수이기 때문에 한국인의 얼굴 검출 문제에는 적합하지 않다.

본 논문에서는 이를 해결하기 위해 한국 연예인을 대상으로 한 데이터 셋을 만들고, 이를 이용하여 이전에 학습된 합성곱 신경망 모델에 대한 실험 분석을 통해 한국 데이터에 대해 이전에 학습된 합성곱 신경망을 효율적으로 이용하여 결과를 도출할 수 있는 방법을 연구한다. 구성은 다음과 같다. 2장에서는 얼굴 검출에서 많이 쓰이는 연구들을 살펴본다. 3장에서는 수집한 한국 연예인 데이터와 미디어에서의 인물 검출 방법, 그리고 실험 결과를 분석하고, 이를 이용한 Django 기반의 얼굴 검츨 서비스를 이야기 한다. 마지막 6장에서는 결론과 추후 개선 사항에 대해 논의한다.

**2. 관련 연구**

본 논문에서는 얼굴 인식을 위한 방법으로 OpenFace[3]와 FaceNet[4]을 사용하였다.

**2.1 OpenFace**

OpenFace[3]는 dlib[5]의 real-time pose estimation 방법과 OpenCV를 사용하여 미리 학습시킨 모델을 이용해서 얼굴의 영역을 찾고 OpenCV의 affine 변환을 이용하여 모든 사진의 눈과 입의 위치를 맞추어 신경망 학습에 용이한 형태로 변환한다.

**2.2 FaceNet**

FaceNet[4]은 MS-Celeb-1M[4] 데이터를 Inception ResNet v1[6] 모델을 이용하여 학습하며, 손실 함수는 ‘Deep Face Recognition’ [7]에서 소개한 Triplet Loss 방법을 이용하였다. 본 논문에서는 이 중 분류기만을 학습하여 결과를 도출한다.

**3. 실험**

**3.1 데이터**

한국 연예인을 대상으로 가수, 배우, 개그맨 등 다양한 직종의 인물 1,000명에 대한 사진을 약 12만장 가량 수집하였다. 각 인물마다 많게는 180장에서 적게는 50장 정도를 가지고 있다. 구글에 있는 사진들을 crawler를 이용하여 수집하고 OpenFace[1]에 있는 방법을 이용하여 얼굴을 검출하여 수집하였다.

3.2 학습 데이터의 개수에 따른 성능 측정

효율적인 얼굴 검출을 위한 학습을 위해 첫번째 실험으로 분류기의 학습을 위해 각 인물 당 어느 정도의 사진 개수가 필요한지 실험을 진행하였다. 실험은 테스트 데이터의 개수를 전체 데이터의 20%로 고정하여 테스트하고 학습 데이터의 개수만 변화시켜가며 실험을 진행하였다. 인물의 수는 데이터의 개수가 가장 많은(150장 이상) 상위 255명이고, 사진의 크기는 160 x160 이다. 표 1. 에 따라 학습 데이터의 개수가 증가함에 따라서 성능이 올라가는 것을 확인할 수 있으며 이를 통해 인물 당 학습 데이터의 수가 약 90~100장 이상이라면 효율적으로 분류기를 학습할 수 있다는 것을 확인했다.

**3.3 사진의 크기에 따른 성능 측정**

효과적인 얼굴 검출을 위한 학습을 위해 두번째 실험으로 어느 정도까지의 얼굴 크기가 효과적인 결과 값을 도출하는 지 확인하기 위한 실험을 진행하였다. 이전에 학습된 모델의 입력 값이 160x160 크기의 얼굴이므로 160x160 크기를 원본으로 설정하고 이를 10%씩 감소시켜가면서 성능을 분석했다. 표 2. 에 따라 80x80까지는 원본보다 성능이 낮긴 하지만, 괜찮은 성능을 나타내는 것을 확인할 수 있었으며, 80x80 이하의 사진에 대해서는 좋지 않은 성능을 보이기 때문에 학습 또는 실험 데이터에 80x80 이상의 사진을 사용해야 좋은 성능을 낼 수 있다는 것을 확인했다.

**3.5 인물의 수에 따른 성능 측정**

이 실험의 가장 큰 목적은 인물 1,000명에 대해 얼마나 정확한 결과를 도출하는 확인하는 것이 아닌, 외국인의 얼굴로 학습한 CNN 모델이 한국인의 얼굴을 이용하여 학습하였을 때, 몇 명 정도까지 효과적으로 결과를 도출하는 지 알아보기 위한 확인이며 255명, 500명, 1,000명을 대상으로 실험하였다.

표 3. 에 따라 1,000명의 인물까지는 성능이 선형적으로 감소하기 때문에 1,000명까지는 인물을 늘려도 분류기의 성능이 나쁘지 않다는 것을 확인하였다. 하지만 더 많은 인물에 대해서는 데이터가 충분하지 않기 때문에 실험으로 확인할 수 없었다.

표 1. 학습 데이터의 개수에 따른 성능 측정 결과

**3.6 한계 및 문제점**

데이터 수집의 문제로 인해 충분한 양의 데이터를 확보하지 못하여 표 1. 에서 보이는 것과 같이 성능이 이미지의 개수 120개에서도 꾸준히 증가하지만 이후에 수렴하는 곳을 찾지 못하는 한계가 있었다. 더 많은 데이터가 수집된다면 정확한 결과를 얻을 수 있을 것이다. 또한 인물의 수에 따른 성능 측정 결과를 도출하는 실험에서 각 인물 당 가지고 있는 사진의 개수가 서로 다르기 때문에 정확한 결과 값을 구할 수 없었다는 한계가 있다. 또한 4.3에서 목표한 몇 명의 인물에 대해 결과를 효과적으로 도출하는 가에 대한 문제도 남아있다. 이전까지는 분류기만을 학습하여 실험하였는데, 합성곱 신경망을 한국 연예인의 얼굴 데이터를 이용해 fine-tuning 하여 실험하였고 결과는 표 3. 과 같다. 인물의 수에 따라 실험하였으며 정답률이 1~5퍼센트 가량 상승하였다. 인물의 수가 낮은 경우에는 데이터의 수가 많아서 fine-tuning이 잘 되었기 때문에 높은 성능 향상을 보였다. 이는 합성곱 신경망이 이전에 학습된 데이터인 외국인 얼굴과 fine-tuning 시 사용된 데이터인 한국인 얼굴의 서로 다른 특징을 학습한 것을 의미하며, 합성곱 신경망이 이용되는 곳에 따라 이용 되는 곳에 맞는 데이터로 학습시키는 것이 좋은 성능을 낼 것이라는 것을 알 수 있다.

**3.7 Django를 이용한 얼굴 검출 웹 서비스**

이번 프로젝트에서는 이전에 진행한 실험 결과와 더불어 Django를 이용하여 실험에 사용한 Tensorflow를 웹 상에서 구현하는 것을 목표로 하였고 언어는 Python을 이용하였다.

구조는 html을 이용하여 사용자에게 얼굴 영역의 사진을 입력으로 받아서 서버로 전달하면 서버에서 구현한 Tensorflow로 만들어진 얼굴 검출을 실행한 후 결과를 다시 사용자에게 출력하는 간단한 구조이다. 웹 프로그래밍이 익숙하지 않아 예정한 시간보다 오래 걸려 완성도가 떨어지는 **표 2**. 사진의 해상도에 따른 성능 측정 결과

것이 단점이고, 입력을 얼굴 영역으로 받아야한다는 것도 단점이기 때문에 추후에 이를 보완하도록 하겠다.

**4. 결론 및 추후 연구**

얼굴 검출 문제를 해결하는데 필요한 합성곱 신경망 모델을 효율적으로 학습하기 위한 방법을 여러 기준의 실험을 통해 확인했다. 그 결과 기존에 있는 합성곱 신경망 모델을 사용해도 1,000명의 얼굴을 검출하는 데는 큰 문제가 없었으나, 하지만 합성곱 신경망을 하였을 때 더 좋은 결과가 나왔다. 얼굴 영역을 검출하는 웹 사이트 구조는 그림 2. 와 같다. 사용자에게 얼굴 영역 이미지를 입력을 받아서 서버에 보내면 서버에서는 이를 구현한 코드를 이용해서 처리한 후 결과와 이미지를 다시 사용자에게 보여준다. 단점은 위에서 말했던 단점이 있기에 얼굴 영역을 넣지 않아도 얼굴 영역을 찾아주는 기능을 추가할 예정이다. 또한 속도를 더 가속할 수 있는 방안도 모색 중이다.

|  |  |  |
| --- | --- | --- |
| 인물의  수 | 분류기 Fine-tuning시 정답률 | 분류기 + 특징 추출기  Fine-tuning시 정답률 |
| 255 | 0.939 | 0.990 |
| 500 | 0.925 | 0.984 |
| 1000 | 0.903 | 0.913 |

**표 3.** 분류기, 특징 추출기 Fine-tuning에 따른 성능 측정 결과

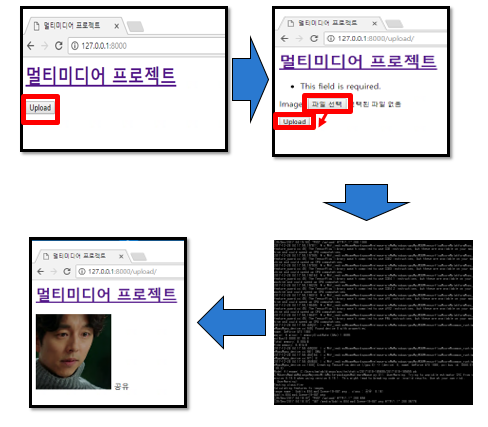


그림 2. 얼굴 영역을 검출하는 웹 사이트 구조

**참 고 문 헌**

[1] D. Yi, Z. Lei, S. Liao and S. Z. Li. “Learning Face Representation from Scratch,” *arXiv preprint arXiv:1411.7923*, 2014.

[2] Y. Guo, L. Zhang, Y. Hu, X. He and J. Gao. “Ms-celeb-1m: A Dataset and Benchmark for Large-scale Face Recognition.,” in *Proceedings of European Conference on Computer Vision,* pp. 87–102. 2016.

[3] B. Amos, B. Ludwiczuk and M. Satyanarayanan, Openface: A General-purpose Face Recognition Library with Mobile Applications, *Tech. Rep., CMU-CS-16-118, CMU School of Computer Science*, 2016.

[4] F. Schroff, D. Kalenichenko and J. Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering,”in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 815-823, 2015.

[5] D. King, dlib Library. [online]. Available: <https://dlib.net> (Released Sep 17, 2017.)

[6] C. Szegedy, S. Ioffe, V. Vanhoucke and A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.,” in *Proceedings of Association for the Advancement of Artificial Intelligence*, pp. 4278-4284, 2017.

[7] O.M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep Face Recognition.,” *British Machine Vision Conference*, vol. 1, no. 3, pp. 6, 2015.

[8] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, “Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks,” *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499-1503, 2016.