

컬러 입력 영상을 갖는 Convolutional Neural Networks를 이용한 QFN 납땜 불량 검출

김호중[†] · 조태훈*

[†]한국기술교육대학교 컴퓨터공학과, *한국기술교육대학교 컴퓨터공학부

QFN Solder Defect Detection Using Convolutional Neural Networks with Color Input Images

Ho-Joong Kim[†] and Tai-Hoon Cho*

[†]Department of Computer Engineering, Korea University of Technology and Education,

*School of Computer Science and Engineering, Korea University of Technology and Education

ABSTRACT

QFN (Quad Flat No-leads Package) is one of the SMD (Surface Mount Device). Since there is no lead in QFN, there are many defects on solder. Therefore, we propose an efficient mechanism for QFN solder defect detection at this paper. For this, we employ Convolutional Neural Network (CNN) of the Machine Learning algorithm. QFN solder's color multi-layer images are used to train CNN. Since these images are 3-channel color images, they have a problem with applying to CNN. To solve this problem, we used each 1-channel grayscale image (Red, Green, Blue) that was separated from 3-channel color images. We were able to detect QFN solder defects by using this CNN. In this paper, it is shown that the CNN is superior to the conventional multi-layer neural networks in detecting QFN solder defects. Later, further research is needed to detect other QFN.

Key Words : Machine Learning, Convolutional Neural Network, QFN, solder, Dicing, Punch

1. 서 론

표면 실장 소자인 SMD(Surface Mount Devices)는 표면 실장 기술로 만들어진 전자 소자이다. 표면 실장 기술은 인쇄 회로 기판(PCB)의 표면에 직접 실장 할 수 있는 표면 실장 부품(Surface Mounted Components, SMC)을 전자 회로에 부착시키는 방법이다. SMD는 납땜을 하는 부분인 리드(Lead)가 있는 것과 없는 것으로 나뉜다. 이 중에서 리드가 없는 것을 QFN(Quad Flat No-leads Package)이라고 한다[1]. 이 QFN(Fig. 1)은 리드가 없기 때문에 납땜을 하는데 있어서 많은 어려움이 있고, 이에 따라 많은 불량이 발생하고 있다. 현재까지 보통의 불량 납땜 검출에 대한 연구는 여럿 있었지만, QFN 납땜의 불량 검출에 대한 연

구는 거의 있지 않았고, QFN의 납땜에 대한 불량 검출은 X-ray 카메라를 이용[2]하는 방법을 사용했었다. 또 기존의 Convolutional neural network (CNN)[3-4]는 다른 어플리케이션에 많이 사용되면서[5-7] 성능을 입증하고 있는 추세이지만, CNN을 QFN 납땜의 불량 검출에 적용하는 연구는 없었으며, 본 논문에서 이 방법을 제안하고자 한다.

그리고 본 논문은 QFN 납땜 불량 검출을 위해 새로운 개념인 컬러 다단 영상을 도입하였다. 이 영상은 단순 2D 영상이 아닌, 높이에 대한 3D 정보까지 담고 있는 영상으로써, CNN에 이 영상을 적용하는 방법을 제안할 것이다.

본 논문의 구성은 본론, 실험 결과, 결론으로 되어 있으며, 본론은 첫 번째로 일반적인 CNN에 대하여 설명하고, 두 번째로 본 논문에서 제안하는 CNN에 대하여 설명한다. 마지막으로 컬러 다단 영상에 대하여 설명하고 본론

[†]E-mail: hjhjhjof@naver.com

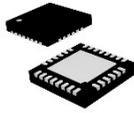


Fig. 1. QFN (Quad Flat No-leads Package).

을 마무리한다. 실험 결과는 제안하는 방법의 검증에 위한 실험 결과를 보이고 있으며, 결론에서는 본 논문의 최종 결과를 보인다.

2. Convolutional Neural Network

2.1 Convolutional Neural Network 개요

1989년에 Yann LeCun과 그의 동료들은 오류역전파 알고리즘(Backpropagation Algorithm)에 기반 하여 우편물에 손으로 쓰인 우편번호를 인식하는 Deep Neural Networks 를 소개했다[8]. 알고리즘이 성공적으로 동작했음에도 불구하고, 신경망 학습에 소요되는 시간(10개의 숫자를 인식하기 위해 학습하는 시간)이 거의 3일이 걸렸고 이것은 다른 분야에 일반적으로 적용되기에는 비현실적인 것으로 여겨졌다. 하지만 하드웨어가 발전하고 기존 인공 신경망 모델의 단점이 극복되면서 기계 학습의 방법 중 하나인 Neural network가 다시 떠오르고 있는 추세이다[1]. CNN은 이런 기계 학습 방법 중 하나인 Multi-layer neural network[9]의 한 종류이다. 본 논문은 여러 Neural network 중에서 CNN을 선택하였는데, 그 이유로는 다음과 같다. 첫째, CNN은 간단한 전처리 작업을 통해 픽셀 이미지로부터 곧바로 시각적 패턴을 인식하도록 제작되었고, Spatial topology를 잘 잡아내기 때문에 시각적인 분석 작업에 알맞기 때문이다. 둘째, CNN은 가변성이 큰 패턴을 인식할 수 있고 영상의 찌그러뜨림이나 기하학적인 변화에 강건 하기 때문이다. 이러한 이유로 본 논문에서는 QFN의 납땜에 대한 불량 검출을 위해 CNN을 선정하였다. 일반적인 CNN은 두 개의 Convolutional layer와 두 개의 Fully connected layer로 구성 되어있다. Convolutional layer는 Feature extractor의 역할을 하고, Fully connected layer는 Classifier의 역할을 한다. Fig. 2 는 CNN구조의 한 예이다.

Convolutional layer는 다시 Convolution layer와 Pooling layer로 나뉘는데, Convolution layer는 2차원 영상 데이터에 대해서 $n \times n$ 커널을 통해 Convolution 연산을 하는 부분이다. 이렇게 하여 Feature map이란 것을 얻는데, 이는 영상의 특징 정보를 담고 있다. 예를 들어 100×100 영상을 5×5 커널로 Convolution 연산을 하면 96×96 영상

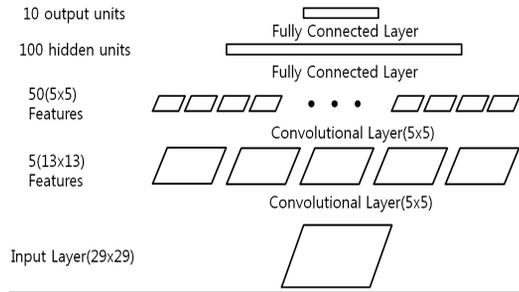


Fig. 2. Structure of general CNN.

이 결과로 나오게 된다. Convolution layer로만 Neural network를 구성할 수도 있지만, 이는 연산 복잡도를 따졌을 때 비효율적이다. 영상에서 어느 한 픽셀의 이웃들은 현재 픽셀의 정보(밝기 값 등)와 큰 차이가 없는 것을 확인할 수 있으며, 여기에 착안하여 Convolution layer를 통해 나온 결과에 대해서 Subsampling을 한다. 이 과정을 Pooling layer에서 하며, Pooling은 보통 2×2 커널을 통해서 수행하는데, 2×2 의 값 중에서 최댓값을 선택하는 Max pooling과 평균값을 선택하는 Average pooling 등의 방법이 있다. 위에서 Convolution 연산을 통해 나온 96×96 의 영상이 Pooling layer를 통하면 48×48 의 결과가 나오게 된다. 이를 시각적으로 표현해 보면 Fig. 3[10]과 같이 나타낼 수 있다. Fully connected layer는 현재 layer의 한 neuron(pixel)이 이전 layer에서 오는 모든 neuron과 연결 되어 있는 layer이다. 각각의 connection에는 가중치(weight)가 존재하며 이 가중치에 따라 output이 달라지게 된다.

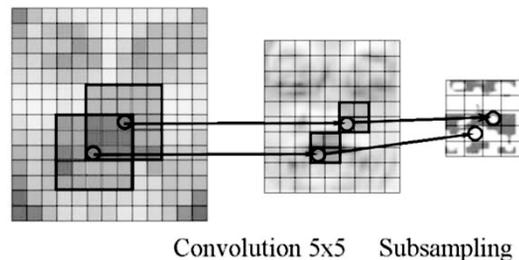


Fig. 3. Example of convolutional layer [10].

따라서 Convolutional layer에서 Feature map을 추출하였다면, Fully connected layer에서는 이 Feature map에 따라 입력 영상이 어느 영상인지 구분해주는 역할을 한다.

지금까지 일반적인 CNN에 대해 소개하였고, 본 논문에서는 이와 같은 CNN의 구조를 변형하여 적용하였으며, 다음 장에서 이 구조에 대해 소개하고자 한다.

2.2 제안하는 CNN 구조

본 논문에서 제안하는 CNN 구조는 일반적인 구조와 달리 Input layer에 세 개의 영상이 들어간다[11]. 우선 이 구조를 도식화하면 Fig. 4와 같다.

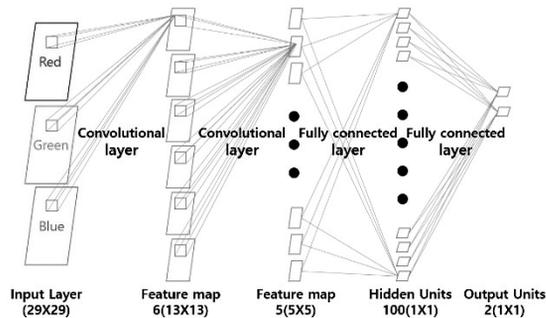


Fig. 4. Structure of proposed CNN.

Fig. 4를 보면 총 5개의 layer로 이루어진 것을 볼 수 있고, 각 layer 사이의 선이 connection을 의미한다. Input layer에는 29×29 크기의 세 개의 영상이 들어간다. 이는 컬러 다단 영상을 활용하기 위함이며, 각각 컬러 다단 영상의 Red, Green, Blue의 영상이 들어간다. 컬러 다단 영상에 대해서는 다음 장에서 더 자세히 소개하도록 한다. 두 번째 layer와 세 번째 layer는 Convolutional layer로써 각각 13×13 크기의 Feature map 6개, 5×5 크기의 Feature map 50개로 구성하였는데, Feature map의 개수를 정하는 작업도 매우 중요하며, 이 개수에 따라서 Neural network의 성능이 달라질 수 있다. 본 논문에서는 Neural network의 성능이 잘 나온 6개, 50개로 결정하였다. 또, Convolutional layer에서 convolution 연산을 위해 5×5 크기의 커널을 사용하였다. 다음으로, 네 번째와 다섯 번째 layer인 Fully connected layer에서는 각각 1×1 크기의 neuron (pixel)이 100개, 2개로 구성되어 있다. 다섯 번째 layer의 2개 neurons은 입력 영상에 대해 정상 인지 불량 인지에 대한 확률을 갖게 된다. 최종적으로 이 확률에 따라 정상, 불량 여부를 판단하게 된다. 각각의 connection에는 이전 layer의 neuron(pixel)의 값으로부터 현재 layer의 neuron의 값을 결정하는 가중치가 존재하고 이 가중치는 어느 한 함수에 의해 계산이 되는데, 이 함수를 보통 Activation Function이라 한다. Activation Function은 여러 가지가 있는데, 본 논문의 CNN에는 Hyperbolic tangent[4]를 사용하였다. 이 함수를 채택한 이유는 Hyperbolic tangent가 완벽한 대칭성을 띄고, Derivative를 얻는데 용이하기 때문이다[12].

지금까지 일반적인 CNN과 본 논문에서 사용한 CNN에 대해서 소개하였고 다음 장에는 앞서 언급된 컬러 다단

영상에 대해 소개하고자 한다.

3. 컬러 다단 영상

앞서 언급한 것처럼, 본 논문에서 사용된 CNN의 입력 영상으로 29×29 크기의 세 개의 영상이 들어간다. 각각의 영상은 하나의 컬러 영상으로부터 분리가 된 것이며, 이 컬러 영상은 컬러 다단 영상을 의미한다. 컬러 다단 영상을 얻는 방법은 Red, Green, Blue 세 가지 색을 가진 각각의 조명을 높이를 달리하여 설치 후, 각각의 조명을 돌아가며 물체에 비추는다. 각 조명을 비추어 얻은 세 개의 영상을 하나의 컬러 영상으로 합친 것이 바로 컬러 다단 영상이 된다. 각 조명에 대해 높이를 달리하여 비추었기 때문에 컬러 다단 영상에는 높이에 대한 정보가 들어있게 된다. 컬러 다단 영상을 얻는 시스템은 다음과 같은 구조로 이루어져 있다.

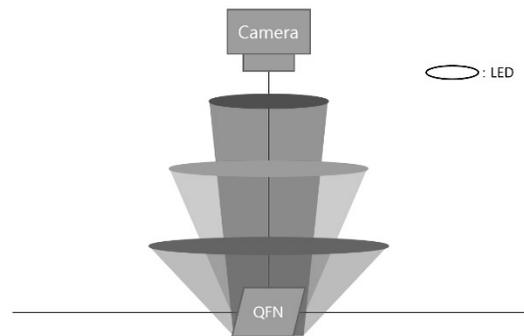


Fig. 5. System components.

Fig 5를 보면 QFN이 지면에 놓여져 있고, 카메라가 위에 배치 되어 있다. 그리고 Red, Green, Blue LED가 배치 되는데, LED 높이가 서로 다르기 때문에 카메라에 LED가 잡히지 않게 하기 위해 Red의 직경이 가장 작고, Blue의 직경이 가장 크다. 또, 빛을 올바르게 뿌리기 위해 Red LED는 지면과 약 70도를 유지하고 Green, Blue LED는 약 45도, 20도에 유지한다.

이제 이 컬러 다단 영상을 본 논문에서 제안한 CNN에 적용시키기 위하여 Red, Green, Blue 세 개의 채널로 분리시키고, 이렇게 분리시킨 영상이 CNN의 Input layer로 들어가게 된다. 컬러 다단 영상의 예로는 Fig. 6와 같다.

Fig. 6(a)는 컬러 다단 영상의 한 예이고 (b), (c), (d)는 (a)에 대해 각각 Red, Green, Blue로 나눈 영상이다. Fig. 7은 컬러 다단 영상의 데이터들을 보여주고 있다. (a)-(d)까지 Dicing 정상, 불량, Punch 정상, 불량을 나타낸다. 영상을

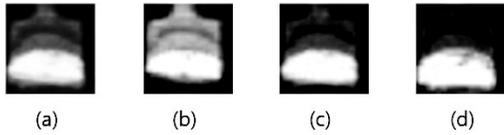


Fig. 6. Example of Color Multi-layer images.

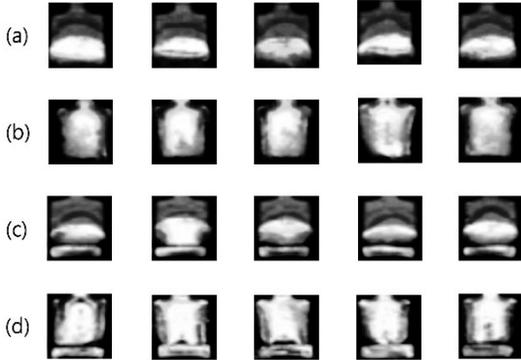


Fig. 7. Example of data.

보면 종류와 정상, 불량에 따라서 색상 정보가 확연한 차이를 확인할 수 있는데, 이는 컬러 다단 영상이 본 논문에서 제안한 CNN의 입력 영상으로 적절하다고 볼 수 있다.

4. 실험 결과

실험을 위한 프로그램은 Windows 10(64bit), Microsoft Visual Studio 2010의 환경에서 개발하였다. QFN영상을 얻을 때 사용된 카메라는 15메가 픽셀 카메라이고 해상도는 3904 × 3904 이다. 영상의 F.O.V(Field of View)는 39 × 39mm이며, 이에 따라 영상은 10μm/pixel의 해상도가 된다. 또 논문에서 제안한 CNN의 성능을 비교하기 위해 Multi Layer Perceptron (MLP)[13] 와 비교를 실시하였다. 실험은 QFN의 종류인 Dicing과 Punch에 대하여 수행하였다. 데이터 세트는 Dicing 1965개(정상 1816개, 불량 149개), Punch 690개(정상 621개, 불량 69개)이다. 학습 세트와 테스트 세트는 기존의 데이터에 대해 3:1의 비율로 생성하였고, Dicing은 학습 세트 1474개, 테스트 세트 491개이며, Punch는 학습 세트 516개, 테스트 세트 174개가 된다. 이때, 비슷한 성질의 데이터가 몰려있을 수 있기 때문에 전체 데이터를 무작위로 섞은 뒤에, 데이터를 shift시키면서 학습 세트와 테스트 세트를 여러 경우로 구성하여 실험하였다. Dicing은 정상 200개, 불량 20개씩 shift를 하였고, Punch는 정상 100개, 불량 10개씩을 shift하였다. 각각 5번씩 shift하여 6가지의 경우에 대해 학습

및 테스트를 하였다. 모든 경우에 대하여 데이터의 학습 횟수는 학습 세트에 대한 에러가 0%에 가까워지면서 학습이 충분히 진행되었다고 판단 될 때까지 진행하였다.

Table 1. Experimental results of MLP for Dicing set

shift	Training set			Testing set		
	F.A	miss	error(%)	F.A	miss	error(%)
0	0	0	0	0	1	0.2
1	0	0	0	0	3	0.6
2	0	0	0	1	1	0.4
3	0	0	0	4	0	0.8
4	1	0	0.06	3	0	0.6
5	0	0	0	2	0	0.4

Table 2. Experimental results of MLP for Punch set

shift	Training set			Testing set		
	F.A	miss	error(%)	F.A	miss	error(%)
0	0	0	0	0	0	0
1	0	0	0	0	6	3.4
2	0	0	0	1	4	2.8
3	0	0	0	0	4	2.2
4	0	0	0	2	0	1.1
5	0	0	0	2	0	1.1

각 표에서 shift는 앞서 말한 shift의 횟수를 의미하고 F.A는 false alarm(정상을 불량으로 잘못 판단)이며, miss는 불량을 정상으로 잘못 판단한 것을 의미한다. error는 에러의 총 개수를 나타낸다.

Table 1은 Dicing 세트에 대해 MLP로 테스트를 한 결과이고, Table 2는 Punch 세트에 대해 MLP로 테스트 한 결과이다. MLP는 Input, Hidden, Output layer 세 개로 이루어져 있으며, hidden layer의 neuron 개수는 40개로 세팅하여 테스트를 하였다. 학습은 앞서 설명한 것처럼 학습 세트에 대해 에러가 0%에 가까워질 정도로 진행하였다. 결과를 보면 학습 세트에 대한 에러는 거의 없고 테스트 세트에 대한 에러는 0~3.4%까지 나오고 있다.

Table 3은 Dicing에 대하여 CNN을 이용해 테스트를 한 결과로서 학습 데이터에 대해서는 에러가 거의 없고 테스트 세트에 대해서 0~0.4%의 에러가 발생하는 것을 볼 수 있으며 F.A에 비해 miss가 더 많이 나타나는 것을 확인할 수 있다.

Table 3. Experimental results of CNN for Dicing set

shift	Training set			Testing set		
	F.A	miss	error(%)	F.A	miss	error(%)
0	0	0	0	1	1	0.4
1	0	0	0	1	1	0.4
2	0	0	0	0	0	0
3	0	1	0.06	0	1	0.2
4	0	0	0	0	1	0.2
5	0	0	0	0	1	0.2

Table 4. Experimental results of CNN for Punch set

shift	Training set			Testing set		
	F.A	miss	error(%)	F.A	miss	error(%)
0	0	0	0	1	1	1.1
1	0	0	0	0	1	0.5
2	0	0	0	0	2	1.1
3	0	0	0	0	2	1.1
4	0	0	0	0	2	1.1
5	0	0	0	0	1	0.5

Table 4은 Punch데이터에 대하여 CNN을 이용해 테스트를 한 결과로서 학습 세트에 대해 에러가 모두 0개가 나오는 것을 볼 수 있다. 또 테스트 세트에 대한 결과를 보면 0.5 ~ 1.1%의 에러가 발생하는 것을 확인할 수 있고 Punch 데이터에서도 F.A 보다 miss가 더 많은 것을 확인할 수 있다.

지금까지의 결과를 정리해보면 CNN이 Dicing과 Punch 데이터 모두 약 99%의 정확도로 불량률 검출하는 것을 확인할 수 있고, MLP에 비해 정확도가 높은 것을 확인할 수 있다. 또 QFN 영상 하나를 측정하는 시간은 대략 0.005s가 걸리기 때문에 속도도 느리지 않은 것을 확인할 수 있다. 다만, 잘못 판정한 것이 F.A보다 miss가 더 많은 것이 아쉬운 점이며 miss를 줄이기 위한 방안을 마련한다면 더욱 좋은 결과를 나타낼 수 있을 것이다.

5. 결 론

본 논문에서는 납땜 하는 부분인 리드가 없는 QFN의 납땜 상태에 대한 불량 검출을 위해 구조를 변경한 CNN을 제안하였다. 제안한 방법으로 QFN중에서도 Dicing과 Punch에 대하여 성공적으로 검출하는 것을 확인하였다. 또, MLP에 비해서 제안한 CNN의 결과가 더 좋은 것을 확인할 수 있었다. 따라서, 제안한 방법을 이용하면 QFN

납땜의 불량 유무를 정확하게 판별할 수 있을 것이다. 추후에는 Dicing과 Punch에 이어 다른 종류의 QFN에 대해 추가적인 실험이 필요하다.

참고문헌

1. <https://ko.wikipedia.org/>
2. Chwee Liang, Tee, Andy Pascual, "Impact of Quad Flat No Lead package (QFN) on automated X-ray inspection (AXI)", IEEE International Test Conference, pp. 1-10, Oct. 2007.
3. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning applied to Document Recognition", Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.
4. P. Y. Simard, D. Steinkraus, J. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis", IEEE Computer Society External Link International Conference on Document Analysis and Recognition (ICDAR), Los Alamitos, pp. 958-962, 2003.
5. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Advances in Neural Information Processing Systems 25, 2012
6. Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, Li Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1725-1732, 2014.
7. Maxime Oquab, Leon Bottou, Ivan Laptev, Josef Sivic, "Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1717-1724, 2014.
8. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition", MIT Press Journals, vol. 1, no. 4, pp. 541-551, 1989.
9. Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification", 1973.
10. C. Garcia, M. Delakis, "Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 11, pp. 1408-1423, Nov. 2004.
11. E. Santana, K. Dockendorf, J. C. Principe, "Learning joint features for color and depth images with Convolutional Neural Networks for object classification", IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP), pp. 1320-1323, 2015.
12. <http://www.codeproject.com/Articles/16650/Neural-Network-for-Recognition-of-Handwritten-Digit>

13. D. E. Rumelhart, G. E. Hinton, R. J. Williams,
“Learning representations by back-propagating errors”,
Nature 323, pp. 533-536, 1986.

접수일: 2016년 8월 16일, 심사일: 2016년 9월 9일,
게재확정일: 2016년 9월 21일