

http://dx.doi.org/10.17703/JCCT.2022.8.5.653

JCCT 2022-9-81

## CNN 기반 지문분류 연구 동향

### Research Trends in CNN-based Fingerprint Classification

정혜욱\*

Hye-Wuk Jung\*

**요약** 최근 이미지와 같은 다차원의 복잡한 패턴 인식에 많이 사용하는 CNN(Convolutional Neural Networks)을 적용한 지문분류 방법이 다양하게 연구되고 있다. CNN 기반 지문분류 방법은 일반적으로 특징추출과 분류 단계로 나누어진 두 단계의 과정을 하나로 통합하여 실행할 수 있다. 따라서 CNN 기반 방법은 지문 이미지의 특징을 자동으로 추출할 수 있으므로, 처리 과정을 단축시킬 수 있는 장점이 있다. 또한 불완전하거나 품질이 낮은 지문의 특징을 다양하게 학습할 수 있으므로, 예외 상황의 특징 추출에 대해 유연성이 있다. 본 논문에서는 CNN 기반 지문분류 연구동향을 파악하고, 실험 방법 및 결과 분석을 통해 향후 연구방향에 대해 논의하고자 한다.

**주요어** : 지문분류, 패턴 인식, 특징추출, 합성곱 신경망, 딥러닝

**Abstract** Recently, various researches have been made on a fingerprint classification method using Convolutional Neural Networks (CNN), which is widely used for multidimensional and complex pattern recognition such as images. The CNN-based fingerprint classification method can be executed by integrating the two-step process, which is generally divided into feature extraction and classification steps. Therefore, since the CNN-based methods can automatically extract features of fingerprint images, they have an advantage of shortening the process. In addition, since they can learn various features of incomplete or low-quality fingerprints, they have flexibility for feature extraction in exceptional situations. In this paper, we intend to identify the research trends of CNN-based fingerprint classification and discuss future direction of research through the analysis of experimental methods and results.

**Key words** : Fingerprint Classification, Pattern Recognition, Feature Extraction, CNN, Deep Learning

#### I. 서론

사람의 생체 정보는 보안 시스템의 개인인증이나 신원을 확인하는데 많이 이용하고 있다. 사람의 고유 정보 중 하나인 지문은 스마트 디바이스의 인증, 보안 및 법의학과 같은 분야에 광범위하게 사용되고 있기 때문에 지문인식이나 지문분류 문제를 풀기위한 연구가 오랫동안

동안 진행되고 있다. 지문분류는 초대형 데이터베이스를 처리하는 속도를 높이고 인식의 정확도를 향상시키기 위해 인식 전 단계에서 지문의 클래스별로 패턴을 분류하는 작업이다[1].

지난 수년 동안 지문분류에 대한 연구는 지문의 용선의 패턴에 따라 4개(Arch, Left Loop, Right Loop, Whorl) 또는 5개(Arch, Tented Arch, Left Loop, Right

\*정혜욱, 경기대학교 진성예교양대학 교양학부 조교수  
(제1저자)

접수일: 2022년 7월 31일, 수정완료일: 2022년 8월 25일  
게재확정일: 2022년 9월 8일

Received: July 31, 2022 / Revised: August 25, 2022

Accepted: September 8, 2022

\*Corresponding Author: wukj@kyonggi.ac.kr

Dept. of College of Liberal Arts and Interdisciplinary Studies,  
Kyonggi University, Korea

Loop, Whorl)의 클래스로 분류하는 방법이 제안 되었다[2].

지문분류는 용선의 방향 정보 등 특징 요소를 추출하고 이를 기반으로 유형별 클래스를 분류하는 과정을 거친다. 지문 용선으로 부터 특징 요소를 추출하는 방법은 사람의 전문지식을 기반으로 지문에서 관찰되는 시각적 특징을 추출하는 방법과 추상적인 특징을 자동으로 추출하는 방법으로 나눌 수 있다[3].

사람이 정의한 특징에 기반한 방법은 지문 용선의 방향 값과 특이점 정보 추출을 위한 알고리즘 설계 및 구현 과정이 있다. 반면 자동으로 특징을 추출하는 방법은 이러한 과정을 생략할 수 있기 때문에 처리 과정을 단축시킬 수 있는 장점이 있다. 또한 불완전하거나 품질이 낮은 지문의 특징을 다양하게 학습 할 수 있으므로 예외 상황의 특징 추출에 대해 유연성이 있다[4].

최근에는 지문분류에 사용하는 특징을 명확하게 추출하는 것이 정확도에 많은 영향을 주기 때문에 이미지 패턴인식에 많이 사용되는 CNN(Convolutional Neural Networks)을 적용한 지문분류 방법이 많이 연구되고 있다. CNN은 이미지와 같은 다차원의 복잡한 패턴을 인식하는데 적합하며, 이미지로 부터 특징을 자동으로 추출하고 분류하는 두 단계의 과정을 하나로 통합하여 학습 할 수 있다.

대부분의 CNN은 합성곱 계층(convolution layer), 풀링 계층(pooling layer), 완전 연결 계층(fully connected layer)으로 구성되어 있다. 합성곱 계층에서는 슬라이딩 커널을 적용하여 같은 층 뉴런 간의 가중치와 편향값을 공유하며 특징을 추출하는 역할을 한다. 풀링 계층은 합성곱 계층 뒤에 위치하며, 데이터의 차원을 축소하는 역할을 한다. 풀링의 종류에는 최대값을 취하는 최대 풀링(max pooling), 평균값을 취하는 평균 풀링(average pooling) 등이 있다. 네트워크의 마지막 계층인 완전 연결 계층에서는 각 뉴런이 이전 계층의 모든 뉴런에 연결되고 이미지의 다차원 특징을 1차원 배열 형태로 변환하여 분류 작업까지 수행할 수 있다[5].

본 논문에서는 이러한 CNN을 이용한 지문분류 접근 방법의 주요기능과 분류 성능 개선을 위한 연구동향을 파악 하고, CNN을 기반으로 한 지문분류 방법 및 결과를 비교 분석 후 향후 연구방향을 탐색하였다.

## II. CNN 모델

### 1. AlexNet

AlexNet은 대규모 데이터베이스로 구성된 ImageNet 을 이용한 이미지 인식 대회인 ILSVRC(ImageNet Large Scale Visual Recognition Challenge) 2012에서 우승한 컨볼루션 신경망(CNN) 모델이다.

모델은 그림 1과 같이 8개 계층(5개의 합성곱 계층, 3개의 완전 연결 계층)이 병렬처리 구조로 되어 있다. 첫 번째 계층에서는 227×227×3 크기의 입력 데이터를 11×11×3 크기의 필터와 스트라이드는 4로 합성곱 연산을 하여 96개의 특징맵을 출력 한다. 두 번째 계층부터 다섯 번째 계층의 스트라이드는 모두 1로 하고 출력값은 전 단계의 특징맵과 함께 합성곱 연산을 한다. 그리고 각 계층에서 사용한 필터의 크기와 출력된 특징맵의 개수는 두 번째 계층(5×5×48, 256), 세 번째 계층(3×3×256, 384), 네 번째 계층(3×3×192, 384), 다섯 번째 계층(3×3×192, 256)으로 구성된다. 모든 합성곱 계층은 ReLU 함수로 활성화하고 첫 번째, 두 번째, 다섯 번째 합성곱 계층은 맥스 풀링을 한다. 여섯 번째와 일곱 번째 계층에서는 드롭아웃(dropout)을 적용하고 전 단계의 출력값과 완전연결 되어 각각 4,096개의 뉴런을 출력한다. 마지막 계층에서는 전 단계의 뉴런과 완전연결 되어 1,000개의 뉴런을 출력한다[6].

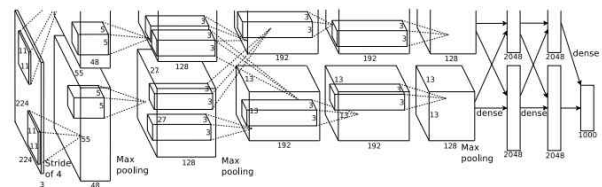


그림 1. AlexNet의 네트워크 구조[6]  
Figure 1. Network structure of AlexNet[6]

### 2. GoogLeNet

파이썬 GoogLeNet은 ILSVRC 2014에서 우승을 차지한 모델로 심층 컨볼루션 신경망 구조에 인셉션 inception) 모듈을 사용했다. 세 개의 합성곱 계층(1×1, 3×3, 5×5)을 병렬로 사용하고, 한 개의 맥스 풀링 계층(3×3)으로 구성된 기본적인 인셉션 모듈에 병목 계층인 1×1 합성곱 계층을 추가하여 연산량을 감소시킴으로써 모델의 성능을 개선했다.

이러한 인셉션 모듈은 신경망 앞부분인 합성곱과 맥스풀링 계층 이후에 9개가 이어져 동작한다. 네트워크

중간에는 두 개의 보조 분류기를 사용하여 신경망 학습이 잘 이루어 질 수 있게 하였고, 마지막 합성곱 계층 다음에 있는 최종 분류단계에서는 연산량을 줄이기 위해 평균 풀링 계층을 사용하여 softmax처리를 하였다. 이와같은 GoogLeNet은 총 22개 계층으로 구성되어 있고 모든 합성곱 계층에는 ReLU 함수를 적용하였다[7].

### 3. VGGNet-VGG16

ILSVRC 2014에서 준우승을 한 VGGNet은 AlexNet과 유사한 형태이지만 합성곱 필터를 3×3의 작은 크기로 사용하여 신경망을 깊게 만든 차이점이 있다. 총 6개의 모델 A(11계층), A-LRN(11계층), B(13계층), C(16계층), D(16계층), E(19계층)의 성능을 비교한 결과 D모델에 해당하는 16계층인 VGG16이 가장 좋은 성능을 나타내었다. VGG16의 구조는 3×3 필터로 합성곱 연산을 두 번 하거나 세 번 진행하고 맥스 풀링은 2×2, 스트라이드는 2로 설정하였다. 결과적으로 특징 맵의 크기가 절반으로 줄어들고 깊이는 두 배로 증가시켰다[8].

### 4. ResNet

ResNet은 ILSVRC 2015에서 우승한 총 152개의 계층으로 구성되어 있는 심층 신경망 모델이다. 이 모델은 합성곱 계층과 항등 매핑(identity mapping) 경로로 구성되어 있는 그림 2와 같은 잔차블록(residual block)을 사용하였다.

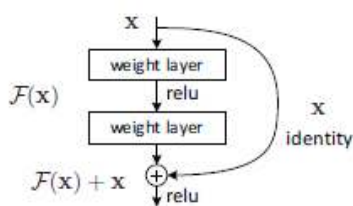


그림 2. ResNet의 잔차블록[9]  
 Figure 2. Residual Block of ResNet[9]

입력값 X를 합성곱 출력값에 더해 입력과 출력 사이의 차이인 잔차 학습(residual learning)을 할 수 있게 하고 50계층이상 부터는 GoogLeNet과 같이 1×1 크기의 합성곱 계층으로 채널의 수를 줄이는 병목 계층을 추가하여 계산의 효율을 높였다. 모델 훈련을 위해서는 합성곱 계층 뒤에 배치 정규화(batch normalization)를 실행한 후 ReLU 함수를 사용하였다[9].

### III. CNN 기반 지문분류 연구동향

Peralta[10] 등은 지문 이미지의 복잡한 패턴을 효과적으로 추출하기 위해 대표적인 CNN 구조인 AlexNet에서 합성곱 계층과 풀링 계층의 위치를 변형한 Caffe Net 모델을 지문분류 문제에 알맞게 수정하여 사용했다. 네트워크는 확률적 경사 하강법인 SGD(Stochastic Gradient Descent)로 학습 하였고, 그림 3과 같이 합성곱 계층 4개(11×11×48, 5×5×128, 3×3×192, 3×3×128), 3×3 크기의 최대 풀링 계층 3개, 완전 연결 계층 3개(2,096, 256, 5)로 구성하였다. 뉴런 수는 총 51,529개로 줄여 검색 공간을 단순하게 설계하고 학습을 위한 입력값인 회색조 227×227 픽셀 크기의 지문 이미지로부터 전역적(global) 평균 값을 계산 후 입력 평균이 0이 되도록 하는 과정을 통해 네트워크 학습과정의 수렴을 가속화 하였다. 이 방법은 수동으로 설계된 방법보다 지문의 기본 구조를 잘 학습하고 높은 정확도를 나타냈으며, 런타임(runtime) 부분에도 효율적인 결과를 나타내었다.

Layer type	Size	Stride	Grouping	Activation
Convolutional	11 × 11 × 48	4	-	ReLU
Pooling	3 × 3	2	-	-
Convolutional	5 × 5 × 128	1	2	ReLU
Pooling	3 × 3	2	-	-
Convolutional	3 × 3 × 192	1	-	ReLU
Convolutional	3 × 3 × 128	1	2	ReLU
Pooling	3 × 3	2	-	-
Fully connected	2096	-	-	ReLU+Dropout
Fully connected	256	-	-	ReLU+Dropout
Fully connected	5	-	-	SoftMax

그림 3. Peralta 등이 제안한 CNN 구조[10]  
 Figure 3. CNN structure proposed by Peralta et al.[10]

Ge[11] 등은 CNN을 기반으로 지문의 추상적(abstract)이며 전역적(global)인 특징을 자동으로 추출하고 지문 분류를 수행하는 방법을 제안 했다. 지문 분류를 위한 Res-FingerNet은 그림 4와 같이 4개의 합성곱 계층, 최대 풀링 계층, 완전 연결 계층(fc5), softmax 계층으로 되어있다. 첫 번째 합성 계층은 커널 크기가 7×7이고 스트라이드는 2이다. 나머지 세 개의 합성곱 계층의 커널 크기는 3×3, 스트라이드는 1이며 모든 풀링 레이어의 커널 크기는 2×2로 설정했다. 256×256 픽셀 크기의 지문 이미지를 입력받은 합성곱 계층에서는 잔차(residual) 매핑 과정을 통한 연산과 풀링 계층에서의 잡음 활성화를 최소화하기 위해 최대 풀링을 선택하여 최적의 출력값을 산출한다. 이 과정에서 입력 이미지의

차원을 줄이고 추상화된 특징맵이 생성되며 완전 연결 계층(fc)에 연결된 후 320 차원의 벡터를 출력한다. 네트워크의 마지막 계층에서는 활성화 함수 softmax를 이용하고 분류의 정확도를 높이기 위해 중심 손실(center loss)을 활용하여 지문의 클래스간 특징을 잘 식별 할 수 있게 하였다.

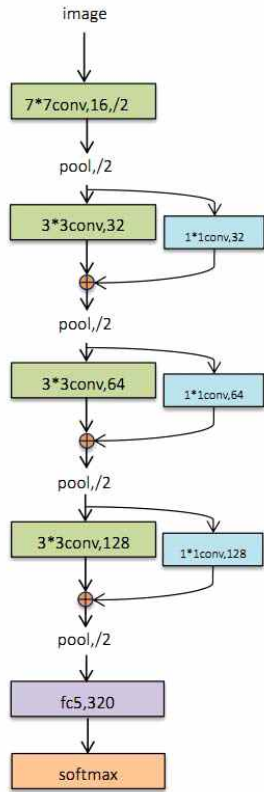


그림 4. Res-FingerNet의 구조[11]  
Figure 4. Structure of Res-FingerNet[11]

Michelsanti[12]등은 AlexNet과 유사한 CNN 모델 중 처리 속도가 빠른 VGG-F와 정확도가 높은 VGG-S를 사용했다. 지문의 클래스는 4개(Arch, Left Loop, Right Loop, Whorl)의 유형으로 나누고, 네트워크의 학습을 위해 입력되는 지문 이미지는 회전, 확대 및 축소, 가우시안 노이즈와 그레이 레벨 매핑을 적용하여 수정하고 크기는 이미지의 중앙 400×400 픽셀 부분에서 자른 후 224×224 픽셀로 조정하였다. 각 모델은 그림 5와 같이 5개의 합성곱 계층, 3개의 최대 풀링 계층, softmax 계층을 포함한 총 19개의 계층으로 구성되어 있으며 ReLU 활성화 함수를 사용하였다. 네트워크 학습을 가속화하기 위해 배치 정규화 계층을 추가하고 두 개의

드롭아웃 계층을 사용하여 과적합을 감소시켰다. 이러한 과정을 통해 전처리 단계가 생략된 지문분류 작업에서 처리시간이 단축되고 정확도가 향상되는 결과를 나타냈다.

VGG-F										
layer type	0	1	2	3	4	5	6	7	8	9
support	-	11	1	3	5	1	3	3	1	3
filt dim	-	3	-	-	64	-	-	256	-	256
num filts	-	64	-	-	256	-	-	256	-	256
stride	-	4	1	2	1	1	2	1	1	1
pad	-	0	0	0×1×0×1	2	0	0	1	0	1
layer type	10	11	12	13	14	15	16	17	18	19
support	1	3	1	3	6	1	1	1	1	-
filt dim	-	256	-	-	256	-	4096	-	4096	-
num filts	-	256	-	-	4096	-	4096	-	1000	-
stride	1	1	1	2	1	1	1	1	1	1
pad	0	1	0	0	0	0	0	0	0	0

VGG-S										
layer type	0	1	2	3	4	5	6	7	8	9
support	-	7	1	3	5	1	2	3	1	3
filt dim	-	3	-	-	96	-	-	256	-	512
num filts	-	96	-	-	256	-	-	512	-	512
stride	-	2	1	2	1	1	2	1	1	1
pad	-	0	0	0×2×0×2	0	0	0×1×0×1	1	0	1
layer type	10	11	12	13	14	15	16	17	18	19
support	1	3	1	3	6	1	1	1	1	-
filt dim	-	512	-	-	512	-	4096	-	4096	-
num filts	-	512	-	-	4096	-	4096	-	1000	-
stride	1	1	1	3	1	1	1	1	1	1
pad	0	1	0	0×1×0×1	0	0	0	0	0	0

그림 5. VGG-F와 VGG-S 모델[12]  
Figure 5. VGG-F and VGG-S models[12]

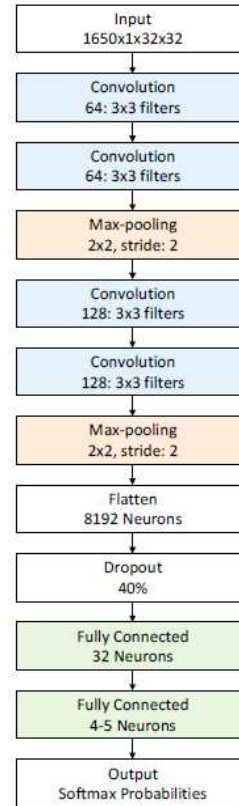


그림 6. Shrein이 제안한 CNN 구조[13]  
Figure 6. CNN structure proposed by Shrein[13]

Shrein[13]은 이미지 전처리 기법인 feature-wise를 적용하여 학습 이미지 픽셀의 평균과 표준편차를 구하고 이를 이용해 표준화하는 방법으로 CNN에 입력되는 이미지의 차원을 256배 줄였다. CNN 구현은 케라스(keras) 라이브러리를 사용하였고 구성방식은 그림 6과 같다. 각 4개의 합성곱 계층에서는 입력 또는 이전 계층의 출력 이미지와 3×3 필터로 구성되고 최대 풀링 계층은 2×2 필터에서 스트라이드를 2로 선택 후 출력값에 대한 샘플링 작업을 하며 활성화 함수는 ReLU를 사용한다. Flatten에서는 이전 계층의 출력값이 벡터화 및 모든 뉴런이 연결되며 Dropout을 사용하여 새로운 모델을 정규화하는 작업이 이루어진다. 완전 연결 계층에서는 128개의 3×3 맵으로 32개 뉴런을 형성 후 총 262,176개의 가중치를 생성한다. 이러한 방법을 통해 이미지 전처리 작업으로 차원을 축소하여 CNN의 훈련시간을 단축시키며 지문 이미지 분류에 대해 높은 정확도를 나타내는 결과를 보였다.

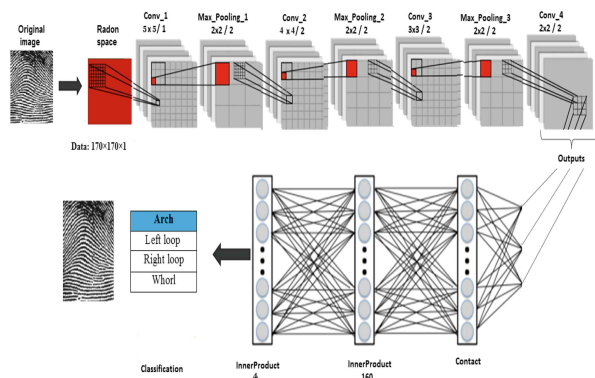


그림 7. El Hamdi 등이 제안한 CNN 구조[14]  
 Figure 7. CNN structure proposed by El Hamdi et al.[14]

El Hamdi[14]등은 이차원 이미지를 벡터값으로 변환하는 RT(Radon Transform)를 확장한 CRT(Conic Radon Transform)를 원본 회색조의 원본 이미지에 적용한 후 CNN의 입력 이미지(170×170)로 사용하였다. 합성 및 풀링 계층에서는 이미지의 회전과 같은 기하학적 변형에 불변하는 지문 용선의 특징을 추출하고 완전 연결 계층에서는 추출된 특징에 가중치를 부여한 후 결합하여 분류 작업을 수행한다. 그림 7과 같이 네트워크는 서로 다른 크기(5×5, 4×4, 3×3, 2×2)의 필터와 1 또는 2의 스트라이드로 4개의 합성곱 계층과 3개의 최대 풀링 계층(2×2)을 구성하였으며 과적합 문제를 방지하기 위해

dropout을 사용하였다. 출력 활성화는 softmax를 사용하여 지문 이미지의 4개 클래스 중 하나로 분류하였다. 이러한 접근 방식은 지문 분류 모델의 학습 및 테스트 시간을 단축시키며 정확하게 분류하는 결과를 보였다.

Wu[15]등은 지문 분류를 위한 클래스를 Arch(A), Whorl(W), Double Whorl(DW), Ulnar Loop(UL), Radial Loop(RL), Peacock Eye(PE)와 같이 6개로 나누고 Caffenet을 변형한 FCTP-Net 모델을 제안했다. 지문 이미지는 256×256 픽셀의 크기로 분할하여 입력 데이터로 사용하고 네트워크는 합성곱 계층 4개(B1~B4), 최대 풀링 계층 3개, norm 계층 2개, 완전 연결 계층 3개(FC5~FC7)인 총 13계층으로 그림 8과 같이 구성하고 활성화 함수는 ReLU를 사용하였다. 블록 B1~B4의 합성곱 필터의 크기는 각각 11×11, 5×5, 3×3, 3×3로 각 풀링 계층은 3×3 크기의 필터와 스트라이드를 1로 설정하여 연산을 수행하였고 합성계층은 풀링 계층과 교대로 계산되며 3개의 완전 연결 계층은 각각 4,096, 4,096, 6개 이다. 이러한 FCTP-Net은 전처리 작업을 하지 않은 지문 이미지의 6개 클래스에 대해 30번의 반복적인 실험을 한 결과 효과적인 지문 분류 결과를 나타냈다.

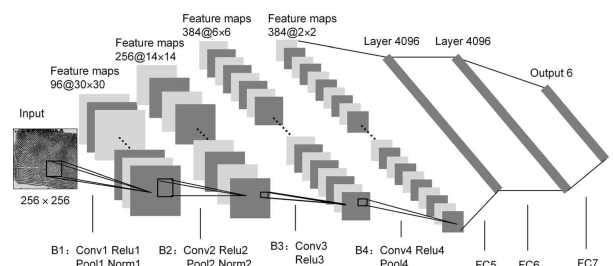


그림 8. FCTP-Net 모델의 구조[15]  
 Figure 8. Structure of FCTP-Net model[15]

Zia[16]등은 Peralta[10] 등이 제안한 CNN과 베이지안 모델의 불확실성을 이용한 효율적인 지문분류 방법을 제안했다. 입력 지문 이미지는 254×254 픽셀의 크기이고 네트워크 구성은 그림 9와 같이 3×3 크기로 3~6개의 합성곱 계층 연산이 수행되며 2×2 크기의 최대 풀링 계층은 합성곱 계층 후에 적용된다. 활성화 함수 ReLU와 LReLU는 2개의 완전 연결 계층 후에 테스트되며 지문의 각 클래스에 대한 사후 확률 산출을 위해 dropout을 사용한다. 각 클래스의 신뢰도와 불확실성을 판단하기 위해 확률의 평균과 분산이 사용되었고, 휴리스틱



(heuristic) 기능을 이용하여 5개 클래스에 대해 최종 분류를 수행한 결과 다른 방법에 비해 일반화되고 정확한 분류 성능을 보였다.

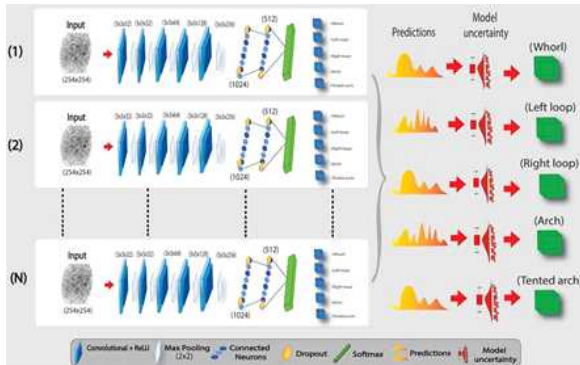


그림 9. Zia 등이 제안한 CNN 구조[16]  
Figure 9. CNN structure proposed by Zia et al.[16]

Nur-A-Alam[17] 등은 가보 필터(gabor filters)와 CNN을 이용하여 추출한 지문의 특징을 융합하여 5개의 클래스(Arch, Tented Arch, Left Loop, Right Loop, Whorl)로 분류하는 방법을 제안했다. 227×227 크기 지문이미지에 가보 필터를 적용하여 STFT(Short Term Fourier Transformation) 방법으로 융선(ridges)과 골(valleys) 정보를 개선 후 방향 및 주파수 값을 이용하여 생성된 지문 이미지의 특징으로 부터 융합 특징 벡터를 추출 및 PCA(Principle Component Analysis)를 적용한 차원을 축소하여 CNN을 학습에 사용하였다.

CNN은 그림 10과 같이 합성곱 계층과 최대 풀링 계층을 조합하여 첫 번째 합성곱 계층은 11×11 크기의 필터 96개와 스트라이드 4, 두 번째 합성곱 계층은 5×5 크기의 필터 256개와 스트라이드 1로 구성하였다. 세 번째부터 다섯 번째 합성곱 계층은 3×3 크기의 필터 384, 384, 256개를 포함하고 stride는 2이다. 모든 합성곱 계층 뒤에는 최대 풀링 레이어를 배치하였고 ReLU 함수를 사용했다.

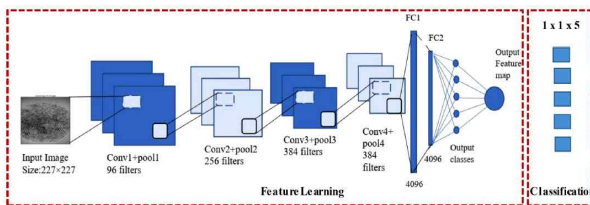


그림 10. Nur-A-Alam 등이 제안한 CNN 구조[17]  
Figure 10. CNN structure proposed by Nur-A-Alam et al.[17]

완전 연결 계층 및 출력 계층에서는 4,096개의 뉴런이 포함되어 있으며 softmax 함수를 적용했다. 이와같이 지문 이미지의 특징 융합과 CNN을 이용하여 지문 분류를 한 결과 정확도를 높이고 거부율을 낮추는 결과를 보였다.

#### IV. CNN 기반 지문분류 방법 및 결과 비교 분석

앞에서 설명한 지문분류 연구들의 성능 비교를 위한 실험방법 및 정확도에 대한 내용은 표 1과 같다.

지문 이미지의 특징추출과 분류를 위한 모델은 기본적인 CNN 구조를 이용하여 설계 하거나 기존에 연구된 베이스라인 모델을 사용하였다. 베이스라인 모델은 네트워크의 학습 속도와 분류 정확도에서 좋은 성능을 보이는 AlexNet을 변형한 CaffeNet을 지문 이미지의 크기와 분류 클래스 수에 맞게 수정하여 사용 하였고 [10][15][16], AlexNet과 유사한 VGG-F와 VGG-S을 이용하여 CNN을 구성하였다[12].

지문 이미지의 특징추출은 별도의 전처리를 하지 않은 상태로 CNN에서 자동으로 처리했지만, 일부 연구에서는 지문 융선의 방향값 정규화나 이미지 개선 작업을 통한 전처리 과정을 포함시켜 시스템의 성능을 개선하였다[13][14][17].

실험을 위한 데이터베이스는 지문 연구분야에 공개적으로 사용되고 있는 NIST(National Institute of Standards and Technology)를 사용했다. NIST-DB4는 512×512 크기의 지문 이미지 2,000쌍이 5개 클래스로 분포되어 있고, 이 중 350개의 지문은 2개의 다중 클래스로 분류되어 있다[18]. NIST-DB300은 수동으로 수집된 888개의 지문을 8비트 회색조의 세 가지 해상도(500, 1,000, 2,000)로 스캔하여 자동화한 지문 이미지 이다[19]. 또한 실험 데이터의 다양성을 고려하여 각 880개의 지문 이미지로 구성되어 있는 FVC(Fingerprint Verification Competition) 데이터베이스 2000, 2002, 2004의 일부를 사용하거나[20-22], SFinGe 소프트웨어를 이용하여 3가지 품질의 지문 이미지 10,000개를 생성하여 사용했다 [23]. 서로 다른 크기의 지문 이미지들은 CNN의 입력 크기를 고려하여 지문 이미지의 크기를 조정하였다.

표 1. 지문분류 연구들의 성능 비교

Table 1. Performance comparison of fingerprint classification studies

저자	베이스라인 모델	전처리	데이터베이스	클래스	정확도(%)
Peralta[10]	CaffeNet	-	SFinGe	5	98.02
			NIST-DB4	5	91.09
Ge[11]	-	-	NIST-DB4	4	97.9
Michelsanti[12]	VGG-F	-	NIST-DB4	4	94.4
	VGG-S	-	NIST-DB4	4	95.05
Shrein[13]	-	방향 벡터의 정규화	NIST-DB4	4	95.45
				5	94.45
El Hamdi[14]	-	CRT	NIST-DB4	4	95.0
Wu[15]	CaffeNet	-	NIST-DB4	6	94.87
				4	92.9
Zia[16]	CaffeNet	-	NIST-DB4	5	96.1
			FVC2002	4	95.8
Nur-A-Alam[17]	-	가보 필터를 이용한 지문이미지 개선	NIST-300 FVC2000 FVC2002 FVC2004	5	99.87

그림 11은 광센서(optical sensor)로 습득된 FVC2002 데이터베이스 DB2의 지문 이미지 중 5개(Arch, Tented Arch, Left Loop, Right Loop, Whorl) 클래스의 예를 보여준다[21].

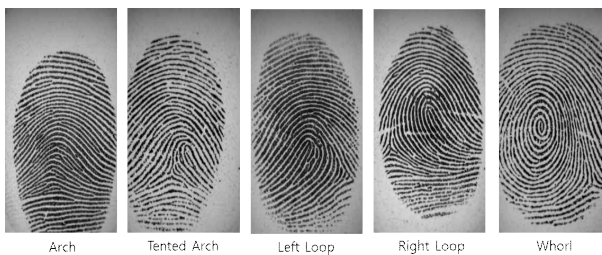


그림 11. 지문 이미지의 5개 클래스[21]

Figure 11. Five classes of fingerprint images[21]

Arch와 Tented Arch의 경우 유사한 패턴을 보이므로 두 개를 하나의 클래스로 간주하여 4개 클래스(Arch, Left Loop, Right Loop, Whorl)로 분류 할 수 있다. 또한 Whorl과 Right Loop를 세분화 하여 6개 클래스(Arch, Whorl, Double Whorl, Ulnar Loop, Radial Loop, Peacock Eye)를 분류 기준으로 사용하기도 했다.

5개 클래스 지문 분류에 대한 정확도는 99.87%의 성능을 보인 Nur-A-Alam[17]의 방법이 가장 우수했다. 이 방법은 전처리 과정과 좋은 품질의 지문 이미지를 선별하는 작업이 포함되어 있다. NIST-DB4에 대한 5개 클래스 분류에서는 별도의 전처리 과정이 없이 다중 클래스의 지문 이미지도 모두 고려한 Zia[16] 등이 제안한

방법이 96.1%로 Peralta[10]나 Shrein[13]에 비해 높은 정확도를 나타내었고, FVC2002를 이용한 4개 클래스 분류에서도 95.8%의 성능을 보였다.

NIST-DB4의 Arch와 Tented Arch 클래스를 병합한 4클래스 분류에서는 97.99%를 보인 Ge[11]의 방법이 가장 높은 정확도를 나타냈다. Shrein[13]은 4개 클래스와 5개 클래스에서 각각 95.45%, 94.45%로 4개 클래스로 분류했을 때 더 좋은 성능을 보였다.

반면 Wu[15]는 4개 클래스 분류 문제에서 Arch와 Left Loop 또는 Right Loop가 오분류 되는 경우를 고려해 6개 클래스로 세분화 하는 방법을 적용하여 4개 클래스(92.9%)에 비해 높은 94.87%의 정확도를 얻었다. Michelsanti[12]은 4개 클래스 분류에서 VGG-F와 VGG-S를 이용하여 각각 94.4%, 95.05%의 정확도를 보였는데 Left Loop와 Right Loop가 오분류 되는 분포가 많았다. El Hamdi[14]은 전처리 작업으로 CNN 모델의 학습 및 테스트 과정을 간소화 시켜 실험한 결과 4개 클래스 분류에 대해 95.0% 정확도를 얻었다.

Peralta[10] 등이 제안한 방법은 SFinGe으로 생성한 다양한 품질의 지문 데이터베이스에 대한 5개 클래스 분류에서 98.02%의 정확도를 보이며 CNN을 이용한 지문분류 방법의 우수한 성능을 제시했다.

Militello[24] 등은 기존 CNN 구성 방식으로 되어 있는 AlexNet, GoogLeNet, ResNet을 이용하여 지문 데이터베이스 PolyU, NIST로 부터 지문을 4개, 5개 그리고 Whorl 클래스를 하위 클래스로 세분화한 8개의 클래스로

분류하는 성능을 비교 및 평가 하였다. 입력 데이터 처리를 위한 전처리 단계에서는 회색조의 지문 이미지를 RGB로 변환하고 각 CNN 구성 방식에 따라 크기를 조정 하였다. 지문 분류 처리 시간을 감소시키며 성능을 유지하기 위해 기존 CNN 구조에서 완전 연결 계층을 4개로 변경하고, 클래스별 출력에 대한 성능 측정을 위해 엔트로피 손실(cross entropy loss)을 사용하였다. 그림 12는 지문을 3가지 유형(4, 5, 8)의 클래스로 분류하기 위해 3개의 CNN과 2개의 다른 데이터베이스를 사용하여 처리하는 과정을 보여준다. 각 CNN의 훈련 방법은 61시간에 해당하는 총 18개의 다른 학습이 반복적으로 이루어진다. 제안 방법은 GUI(Graphical User Interface)를 통해 분석과 테스트를 할 수 있고 모델의 수정 및 분류에 적용할 수 있도록 구현되었다.

표 2는 서로 다른 CNN 모델을 사용해 지문 분류를 수행한 결과에 대한 정밀도(precision)를 보여준다. 지문 이미지의 품질이 좋은 PolyU 데이터베이스는 경우에는 3가지 유형에 대해 모두 유사하게 높은 정밀도를 나타냈다. NIST 데이터베이스의 4, 5클래스 분류에서는 AlexNet과 ResNet이 GoogLeNet보다 높은 정밀도를 나타냈고 8클래스 분류에 대한 정밀도는 AlexNet이 가장 우수한 성능을 보였다.

이와 같은 연구 결과는 원본 지문 이미지의 품질 개선 및 크기 조절, 회전과 같은 기하학적 변형을 고려하여 지문 분류 시스템의 성능 개선 할 수 있는 가능성을 제시했다.

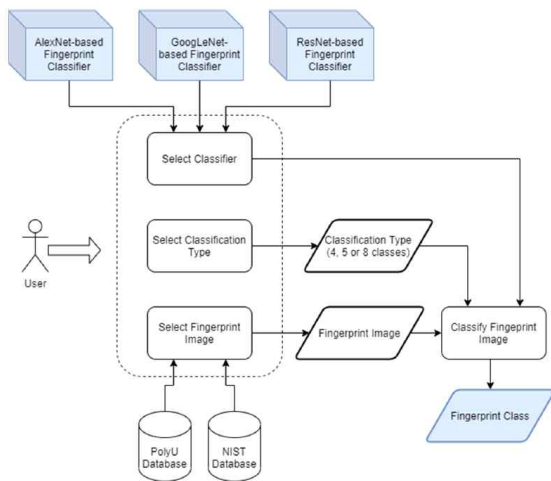


그림 12. CNN 3개와 데이터베이스 2개의 처리과정[24]  
Figure 12. Process of three CNNs and two databases[24]

표 2. 지문 분류 결과의 정밀도  
Table 2. Precision of fingerprint classification results

베이스라인 모델	데이터베이스	클래스	정밀도(%)
AlexNet	NIST-DB4	4	96.85
		5	96.05
		8	93.75
	PolyU	4	99.51
		5	99.79
		8	99.51
GoogLeNet	NIST-DB4	4	93.90
		5	91.55
		8	92.07
	PolyU	4	99.51
		5	99.79
		8	99.58
ResNet	NIST-DB4	4	96.57
		5	95.37
		8	92.71
	PolyU	4	99.65
		5	99.51
		8	99.31

## V. 고찰 및 결론

지문분류에서 중요한 문제 중 하나는 지문 이미지의 품질에 유연하면서 좋은 분류 성능을 가지는 것이다. CNN 기반 지문분류 방법은 자동으로 지문의 특징 추출 및 분류를 할 수 있기 때문에 정확도를 높이고 처리 과정을 단축시키는 장점이 있다. 그러나 잡음이 포함되어 있거나 불완전하게 획득된 지문에 포함된 오류 정보를 특징으로 학습하게 되면 지문 분류의 정확도가 낮아질 수 있다.

CNN 기반 분류기에 입력되는 지문 이미지의 품질을 전처리 작업으로 개선하는 방법도 제안되었지만, 전처리 과정에서 오류가 발생할 수 있고 처리시간의 복잡도가 증가될 수 있다. 지문분류는 클래스를 기준으로 구분하기 때문에 시각적으로 식별하기 어려운 다중 클래스가 존재하는 지문을 정확하게 분류하기에는 한계가 있다. 이러한 문제점을 해결하기 위해 일부 연구에서는 좋은 품질의 지문 이미지만 선별하여 사용하거나 우수한 성능의 지문인식 센서로 습득한 지문 이미지를 이용하여 제한한 방법의 성능을 평가했다. 또한 지문 융선 패턴의 다양성을 고려하여 클래스를 확장하거나 유사한 클래스를 병합하여 분류하는 실험방법도 적용되었다.



CNN과 같은 딥러닝 기술은 사람의 전문적인 지식에 기반한 기술에 비해 자동화된 방법으로 접근할 수 있고 기존에 해결하지 못한 문제점을 개선시킬 수 있기 때문에 많은 분야에 활용되고 있다. 따라서 지문분류 기술에서도 딥러닝의 네트워크 구조를 다양하게 설계하여 시스템의 성능을 향상시키는 연구가 활발히 진행되고 있다. 기존의 특징추출 단계와 분류 단계가 분리된 형태의 시스템을 하나로 병합한 CNN 기반 지문분류 방법은 좋은 성능을 보임을 알 수 있다. 하지만 불완전한 하거나 다중 클래스를 포함하는 지문 분류에 대한 문제와 같이 아직 해결해야 할 부분이 많다.

지문의 용선 정보는 각 클래스의 전역적인 패턴을 판별하기 위해 사용하는 중요한 특징요소이다. 따라서 지문 용선의 일부 정보를 이용하여 전역적 패턴을 추정할 수 있는 방법과 같이 예외적인 데이터 및 클래스에 대한 보다 효율적인 해결 방법이 제시되어야 한다.

## References

- [1] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer, 2009.
- [2] H. W. Jung, L. Seung, "Technical Trend Analysis of Fingerprint Classification," *The Journal of the Korea Contents Association*, Vol. 17, No. 9, pp. 132-144, 2017. DOI: <https://doi.org/10.5392/JKCA.2017.17.09.132>
- [3] H. W. Jung, J. H. Lee, "Noisy and incomplete fingerprint classification using local ridge distribution models," *Pattern Recognition*, Vol. 48, No. 2, pp. 473-484, 2015. DOI: <https://doi.org/10.1016/j.patcog.2014.07.030>
- [4] Zabala-Blanco D, Mora M, Barrientos RJ, Hernández-García R, and Naranjo-Torres J, "Fingerprint Classification through Standard and Weighted Extreme Learning Machines," *Applied Sciences*, Vol. 10, No. 12, 4125, 2020. DOI: <https://doi.org/10.3390/app10124125>
- [5] Y. LeCun, K. Kavukcuoglu, and C. Farnet, "Convolutional networks and applications in vision," *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 253-256, 2010. DOI: <https://doi.org/10.1109/ISCAS.2010.5537907>
- [6] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," In *NIPS*, 2012.
- [7] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9, 2015. DOI: <https://doi.org/10.1109/CVPR.2015.7298594>
- [8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016. DOI: <https://doi.org/10.1109/CVPR.2016.90>
- [10] D. Peralta, I. Triguero, S. Garcia, Y. Saeyns, J.M. Benitez, and F. Herrera, "On the use of convolutional neural networks for robust classification of multiple fingerprint captures," *International Journal of Intelligent Systems*, Vol. 33, pp. 213 - 230, 2018. DOI: <https://doi.org/10.1002/int.21948>
- [11] S. Ge, C. Bai, Y. Liu, Y. Liu, and T. Zhao, "Deep and discriminative feature learning for fingerprint classification," 2017 3rd IEEE International Conference on Computer and Communications (ICCC), pp. 1942-1946, 2017. DOI: <https://doi.org/10.1109/CompComm.2017.8322877>
- [12] D. Michelsanti, A.D. Ene, Y. Guichi, R. Stef, K. Nasrollahi, and T.B. Moeslund, "Fast fingerprint classification with deep neural networks," 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017), pp. 202-209, 2017. DOI: <https://doi.org/10.5220/0006116502020209>
- [13] J. M. Shrein, "Fingerprint classification using convolutional neural networks and ridge orientation images," 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-8, 2017. DOI: <https://doi.org/10.1109/SSCI.2017.8285375>
- [14] D. El Hamdi, I. Elouedi, A. Fathallah, Mai K. Nguyen, and A. Hamouda, "Fingerprint Classification Using Conic Radon Transform and Convolutional Neural Networks," *International Conference on Advanced Concepts for Intelligent Vision Systems, ACIVS 2018*, pp. 402-413, 2018. DOI: [https://doi.org/10.1007/978-3-030-01449-0\\_34](https://doi.org/10.1007/978-3-030-01449-0_34)
- [15] F. Wu, J. Zhu, and X. Guo, "Fingerprint pattern identification and classification approach based on convolutional neural networks," *Neural*

- Computing and Applications, Vol. 32, pp. 5725-5734, 2020. DOI: <https://doi.org/10.1007/s00521-019-04499-w>
- [16]T. Zia, M. Ghafoor, S.A. Tariq, and I.A. Taj, "Robust fingerprint classification with Bayesian convolutional networks," IET Image Process, Vol. 13, pp. 1280 - 1288, 2019. DOI: <https://doi.org/10.1049/IET-IPR.2018.5466>
- [17]Nur-A-Alam, M. Ahsan, M.A. Based, J. Haider, M. Kowalski, "An intelligent system for automatic fingerprint identification using feature fusion by Gabor filter and deep learning," Computers and Electrical Engineering, Vol. 95, 107387, 2021. DOI: <https://doi.org/10.1016/j.compeleceng.2021.107387>
- [18]<https://www.nist.gov/srd/nist-special-database-4>
- [19]<https://www.nist.gov/itl/iad/image-group/nist-special-database-300>
- [20]<http://bias.csr.unibo.it/fvc2000/>
- [21]<http://bias.csr.unibo.it/fvc2002/>
- [22]<http://bias.csr.unibo.it/fvc2004/>
- [23]R. Cappelli, D. Maio, and D. Maltoni, "Synthetic fingerprint-database generation," 2002 International Conference on Pattern Recognition, Vol. 3, pp. 744-747, 2002. DOI: <https://doi.org/10.1109/ICPR.2002.1048096>
- [24]C. Militello, L. Rundo, S. Vitabile, and V. Conti, "Fingerprint Classification Based on Deep Learning Approaches: Experimental Findings and Comparisons," Symmetry 2021, Vol. 13, No. 5, 750, 2021. DOI: <https://doi.org/10.3390/sym13050750>

※ 이 논문은 대한민국 교육부와 한국연구재단의 2017년도 이공학 개인기초연구지원사업(No. 2015R1D1A1A01061064)과 2021년도 인문사회분야 신진연구자지원사업(NR F-2020S1A5A8042850)의 지원을 받아 수행된 연구임.