

http://dx.doi.org/10.17703/JCCT.2018.4.4.393

JCCT 2018-11-52

대형 과수원 사과 분류 시스템

Large orchard apple classification system

김월용*, 신승중**

Weol-Youg. Kim*, Seung Seung-Jung. Shin**

요약 근래 무인화의 발전은 계속되고 있고, AI무인화의 발전은 산업, 복지, 인력등 인력으로 해결해 오던 작업들을 좀더 인력보다 효율적이고 정확하고 신속하게하는 것을 목표로 하고 있다. AI무인화 기술은 다양한 곳에서 발전하고 있는데 이중 많은 산업체나 공장에서 무인화 시스템으로 대대적 전환하는 시점이다. 우리는 이 점을 착안하여 대형 과수원에서 한번에 레일이 쏟아져 들어오는 과일들을 인력이 아닌 인공지능(AI) 핵심 기술중 하나인 Deep Learning 기술을 활용하여 대형 과수원에서 사람이 직접 과일을 분류하지 않아도 자동화 기계가 과일을 종류별, 등급별로 나누어 원산지와 품종 등급별로 나누어 많은 인력을 소비하지 않고 관리자의 감독하에 자동가능한 무인화 과일 분류 기계를 연구하고자 한다. 이러한 무인 자동화 분류 시스템은 인력을 최소한으로 줄여 인건비를 줄이고, 사람이 할 수 있는 실수나 오류들을 최소한으로 줄여 일의 효율성을 증진시킬 수 있도록 하는 것을 목표로 본 연구를 진행하고자 한다.

주요어 : 인공지능, 딥러닝, 알고리즘

Abstract The development of unmanned AI continues, and the development of AI unmanned is aimed at more efficiently, accurately, and speedily the work that has been resolved by manpower such as industry, welfare, and manpower. AI unmanned technology is evolving in various places, and it is time to switch to unmanned systems from many industries and factories. We take this into consideration, and use the Deep Learning technology, which is one of the core technologies of artificial intelligence (AI), not the manpower but the fruits that pour the rails at once in a large orchard. We want to study the unmanned fruit sorting machine that can be operated under manager's supervision without dividing the fruit by type and grade and dividing by country of origin and grade. This unmanned automated classification system aims to reduce the labor cost by minimizing the manpower and to improve the

Key words : AI, Deep Running, algorithm

1. 서론

요즘 과일 가격은 농업 경영체의 수익을 결정하는 중요한 요소 중 하나로 가격의 불안정성은 농업경영체 경영의 위협을 초래하는 한다. 과일 가격의 불안정성

은 소비자들에게 불안한 요소로 작용되며, 자연스레 소비자들의 과일 소비량이 줄어들게 되며, 그에 따라 대형 농장주들은 타격이 크다. 과일 가격의 안정성은 소비자와 농장주 모두 중요하며 과일 가격의 변동을 최대한 줄일 수 있도록 여러 요인들을 찾아 해결해 나

*정회원, 한세대학교 IT융합학과

**중신회원, 한세대학교 ICT디바이스학과(교신저자)

접수일: 2018년 9월 17일, 수정완료일: 2018년 10월 18일

게재확정일: 2018년 10월 27일

Received: September 17, 2018 / Revised: October 18, 2018

Accepted: October 27, 2018

*Corresponding Author: gandalf17@kitech.re.kr

Dept. of IT, Hansei University, Korea

아가고 있을 것이다.

과일가격 변동의 영향을 미치는 중요한 요소들 중 기후의 변화와 인건비에 대한 영향이 크다고 할 수 있다. 날씨 여건에 따른 악천후 발생시 과일 재배량은 현저하게 줄어들게 되고, 생산량이 낮은 과일은 희소 가치성이 올라가 가격 영향에 큰 영향을 미치게 되고, 둘째로 대형 농장 같은 경우 충분한 과일이 생산이 이루어지지 않게 되었을 경우 고정적 인건비에 따른 영향이 과일 가격에 큰 영향을 미치게 된다. 소비자들은 그에 따른 부담비용을 모두 감수 해야 함으로 과일 소비활동이 줄어 농장주들도 큰 타격을 받을 수 있다.

지금까지 인력의 육안으로 해결 해오던 과일의 분류, 판별을 AI Deep Running기술을 이용해 과일의 종류를 판별하고 등급별로 나눌 수 있는 모델을 연구하여 농장주들에게 인력 소모 비용을 줄이고 사람보다 더 정확한 규격대로 과일들을 판별하여 생산성과 정확성을 증진시키고 실질적인 부담비용을 줄여 농장주는 최대한 낮은 가격으로 과일을 판매하고, 소비자또한 최대한 낮은 가격으로 과일을 구매할 수 있도록 하는 것이다.

본 연구는 AI Deep Running기술을 기반으로 하여 실시간으로 들어오는 과일들을 캠으로 촬영하여 실시간으로 촬영된 이미지를 토대로 이미지 분석을 통하여 자동으로 과일의 종류별, 등급별 구분이 가능하여 대형 농장에 인력비용을 최소한으로 줄이고 정확성과 신속성을 올려 농장주에게도 최대한 인력 비용에 대한 부담을 줄이고자 본 모델을 제안 연구하고자 한다.

II. 관련 연구

2.1 딥러닝(Deep Learning) 개요

Deep Learning이란 여러 데이터를 이용해 컴퓨터가 마치 스스로 사람처럼 학습을 할 수 있게 하기 위해 인공 신경망(ANN: artificial neural network)을 기반으로 구축한 한 학습 기술이라 할 수 있다. 즉 사람처럼 컴퓨터가 지식을 터득하고 배울 수 있도록 하는 기술이다. 우리가 알고 있는 Deep Learning 기술로는 최근 이슈화가 된 알파고(AlphaGo)를 예로 들 수 있다. 알파고들 서로 바둑을 학습해 배워 과거에 있었던 바둑 경기들을 스스로 학습하면서 어떤 상황에서는 어떤 수를 두어야 할

지 배워나간 것이다. 사실 Deep Learning 기술은 1980년 처음 등장하였으며 손수 쓴 우편번호를 인식하는데 성공하였다고 한다. 근래의 들어서 Deep Learning 기술이 각광받는 이유는 이전과는 다르게 다양한 데이터가 보급화 되어있어 쉽게 데이터를 수집할 수 있으며, 2012년 글로벌 이미지 인식 경진대회였던 ILSVRC에서 이전의 인공신경망을 개선한 Deep Learning이 이전보다 10%이상을 이미지 인식률을 증가시키며 압도적인 성능 향상을 보여주었고, 이러한 이유로 근래에 Deep Learning 기술은 4차 산업혁명에 대비해 무궁무진하게 발전할 것이다.

2.2 Inception V3 개요

Inception이란 2014년 ILSVRC(이미지 분류 대회)에서 1등을 차지한 기술이며, 기존의 이미지 분류 기술인 CNN기술에 망의 크기를 늘려 성능을 향상시킨 기술이다. 여기서 망의 크기를 늘린다는 것은 단순히 층을 깊게 늘리는 것 뿐만 아니라, 각각의 층에 있는 객체의 수도 늘리는 것을 의미한다. 이전에는 CNN기술에서 망의 크기를 키울수록 파라미터의 수가 증가하게 되어서 망이 과적합에 빠질 가능성이 높았으며, 또 다른 문제로는 망의 크기가 커질수록 연산량이 증가하게 되어 망의 크기를 키워도 최적의 결과를 얻지 못 할 수도 있다. 이를 적절한 망의 크기와 층을 증가시킨 기술이 Google에서 개발한 Inception 기술이다 Inception 모델에 구조의 뼈대가 되는 구조는 NIN(Network In Network)인데, 말그대로 네트워크 속 네트워크를 의미한다. 이를 설계한 설계자는 기존의 CNN의 비선형의 성질을 갖는 특징을 추출하기 어려운 문제를 해결하기 위해 NIN구조를 이용하여 비선형의 성질을 갖는 특징을 추출하였다.

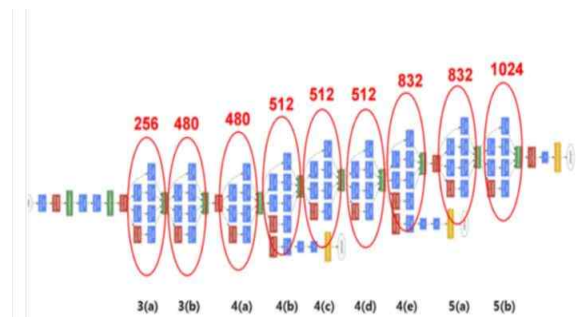


그림 1. Inception 모델 구조.
Figure 1. Inception model structure.

위 그림은 Inception 모델을 구조이며, ㉠ 간색 동그라미위의 숫자는 각 단계에서 얻어지는 특징 Map의 수를, 파란색은 컨볼루션 계층을 빨간색은 맥스-풀링, 노란색은 소프트 맥스 계층을 의미한다. 여기서 컨볼루션 계층은 입력 데이터로부터 특징을 추출하는 역할을 하는데, 필터를 입력받은 데이터의 좌 상단부터 우 하단까지 밀고 나가면서 특징을 추출해 나간다. 여기서 이동하는 간격을 stride라고 하며, 이렇게 필터를 적용해 얻어낸 결과는 최초 입력받은 데이터보다 사이즈가 작아지게 된다. 맥스-풀링 계층에서는 특징을 추출한 Map에서 M*N의 크기로 잘라낸 후 그 안에서 가장 큰 값을 뽑아내 데이터의 크기를 축소 시킨다. 이러한 맥스-풀링 계층을 통과하여 얻는 장점으로는 데이터의 사이즈가 줄어 연산을 적게 할 수 있으며, 데이터가 변화하므로 과적합을 방지 할 수가 있다. 소프트 맥스 계층에서는 입력받은 데이터의 값을 총합이 1인 형태로 바꿈으로서 큰 값을 더욱 두드러지며 작은 값은 더 작아지게 해주는 역할을 한다.

2.3 OpenCV개요

OpenCV(Open Source Computer Vision Library)는 영상처리와 컴퓨터 비전 관련 오픈 소스 라이브러리이다. OpenCV의 컴퓨터 비전 및 영상처리의 라이브러리를 사용하면 응용 프로그램을 빠르고 쉽게 만들 수 있다. OpenCV는 파이썬(Python), C++, C, 매트랩 인터페이스를 갖추고 있고, Linux, Windows 등 다양한 운영체제를 지원한다. Visual Studio 2017 OpenCV를 사용하기 위해서는 매 프로젝트(솔루션(.sln)의 하위 단위(.vcxproj)별로 헤더와 정적 라이브러리들을 링크시켜주는 작업이 필요하다.

2.4 WPF개요

2002년 정식으로 출시된 닷넷(.NET) Framework에서 윈폼(WinForm)이라는 기술이 등장하여 윈도우 응용프로그램 개발을 주도했지만 다양해지면서 미디어, 비디오, 2D 3D그래픽, 애니메이션 등을 다이나믹하게 사용할 수 있기를 원했다. 이러한 부분이 기존의 윈도우 플랫폼들에서 각각이 독립된 기술들로 분산되어 있었고 여러 각각 다른 기술들을 각각 모두 이해해야만 통합된 개발이 가능했다. 결국 통합된 기술이 필요로 하여 만든 기술이 바로 WPF이다.

WPF는 닷넷(.NET) 3.대 버전부터 추가된 기술로 Vista에서 UI를 구현하기 위해 탄생되었으며 Visual Studio

2008에 추가 되었다. 시각적으로 우수한 사용자 환경에서 윈도우 클라이언트 프로그램을 만들 수 있는 차세대 프레젠테이션 시스템이며, WPF를 이용하면 응용프로그램과 브라우저에서 호스팅(Hosting)되는 광범위한 독립 실행 형용용 프로그램을 모두 만들 수 있다.

WPF를 사용하여 Extensible Application Markup Language(XAML), 데이터 바인딩, 애니메이션 레이아웃, 애니메이션, 2D 및 3D 그래픽, 템플릿, 애니메이션, 문서, 미디어, 텍스트, 입력 체계 등의 광범위한 응용프로그램 개발이 가능하다. 또한 클래스를 인스턴스화 한 후 속성을 설정하여 메서드를 호출하고, 이벤트를 처리하는 등의 거의 모든 작업을 Visual Basic, C#, 등의 익숙한 닷넷(.NET) Framework 프로그래밍으로 수행 가능하다. WPF는 대부분 Windows, System, 네임스페이스에 있는 클래스, 인터페이스들을 다룬다. XAML 태그는 UI를 구현하는 데 사용되며 비하인드 코드로 프로그램의 동작을 구현 실행한다. XAML은 UI를 구현하는 데 사용되는 XML 기반 태그 언어이다. 일반적으로 대화 상자, 창, 페이지 및 사용자 정의 컨트롤을 생성하고 이러한 항목을 도형 및 그래픽으로 채우는데 사용된다.

2.5 Anaconda3 개요

TensorFlow는 구글 머신 러닝 인텔리전스(Google Machine Learning Intelligence) 연구소의 구글 브레인 팀(Google Brain Team)의 머신 러닝 인텔리전스 연구 과정에서 개발된 소프트웨어 라이브러리이다. 구글(Google)은 독자적인 Machine Learning Library인 DistBelief에 대한 경험을 바탕으로 TensorFlow를 개발하였고, 2015년 11월 9일에 오픈소스로 공개 되었다. TensorFlow의 주요 기능으로는 다차원 배열 정의, 최적화, 효율적 산술연산을 할 수 있으며, Deep Neural Network와 Machine Learning 프로그래밍을 지원한다. 또한 메모리나 데이터가 자동 관리되기 쉬운 GPU 가속 기능 제공하기 때문에 별도의 코드 없이 TensorFlow가 자동으로 CPU와 GPU에 자원을 할당받아 연산을 한다. 그리고 빅 데이터(Big Data) 처리를 위한 대규모 병렬 컴퓨팅을 지원한다.

TensorFlow는 텐서보드(TensorBoard)를 지원하는데, 텐서보드는 데이터 Flow 그래프 분석 및, 머신 러닝 모델을 시각화함으로 이해도를 높이는 도구이며, 머신러닝 개발 및 튜닝(Tuning)에 유용하게 사용할 수 있다.

TensorFlow는 현재 다양한 곳에서 활용가능 한데, 구

글은 TensorFlow가 스마트폰에서도, 데이터 센터의 수많은 컴퓨터에서도 동작 가능하다고 발표하면서, 어떠한 제약도 받지 않고 유연하게 사용가능한 기술이라고 강조했다. 구글에서 발표한 오픈라이브러리 소프트웨어 이므로 구글 앱 상의 음성인식, 구글 검색, G메일에서 메일을 읽고 상황에 알맞은 예시 답장을 제공하는 '똑똑한 답장 서비스' 등에 적용되었다.

III. 구성도 및 설계

3.1 이미지 분류 구성도

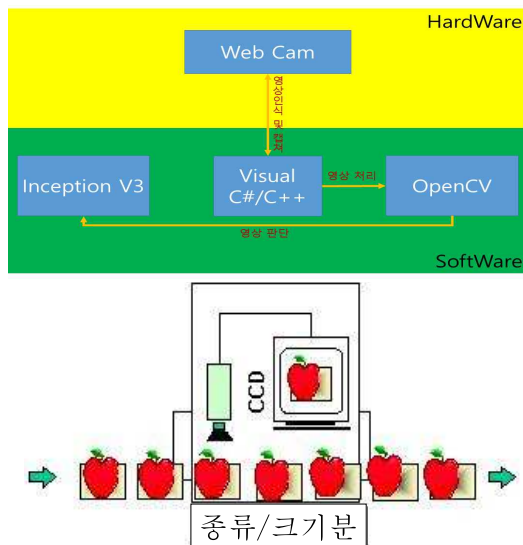


그림 2. 프로젝트 구성도
Figure 2. Project composition diagram.



그림 3. 알고리즘 구현 순서도
Figure 3. Algorithm implementation flowchart

위의 그림을 토대로 전체적은 프로젝트 구성을 보면, 프로그램의 실행과 동시에 USB웹캠으로부터 캡처를 대기하다가 사용자가 결제 버튼을 누르면 캡처가 실행되어 OpenCV로 영상 처리하며, 영상 처리된 객체들로 예측하는 원리이다.

구현 한 알고리즘 순서도인데, 가장 먼저 프로그램이 실행되어 창이 띄워지면, 창의 한 영역에 USB웹캠으로부터 촬영되고 있는 영상이 보여지는데, 캡처되고 있지 않으며 보여지기만 한다. 이후 항시 주기적으로 웹캠 화면 후 OpenCV 라이브러리 프로그램이 실행되는데, 사과외곽선을 검출하여 검출된 객체들을 대상으로 지정한 경로에 JPG형식의 사진 파일로 저장한다. 이후 Inception V3 모듈로 저장된 경로 폴더 내의 jpg형식의 사진들을 예측시킨다. 이 때 사과의 예측도가 85%이상일 경우 해당 파일의 이름들만 텍스트 파일에 저장하게끔 하여 해당 파일이 아니거나 학습시켜 놓은 파일이 아닐 경우에는 그 값이 저장되지 않게 하였다. C#코드로 데이터 바인딩을 통해 텍스트 파일의 값들을 불러들여 창의 한 영역에 파일에 대한 종류, 크기별 등급의 등급 또한 출력되어 컴퓨터가 스스로 어떤 과일인지 예측을 하게 함으로써 파일의 분류가 가능하다.

이번 연구는 이미지 인식에서의 주요 기능 중 하나인 이미지 영역 검출 OpenCV 라이브러리 함수를 사용함으로써 제작한 코드를 통하여 이미지 예측을 수행하여 과일의 종류와 등급을 분별해주는 무인 자동화 시스템이다.

3.2 시스템 실 설계도

위 그림과 같이 사과는 레일을 따라가던 중 캠에 의해서 하나하나 촬영 되고 촬영된 이미지는 이미지 분석을 통해 정상적인 사과가 맞는지 판별하게 된다. 정상적인 사과는 그대로 라인을 통과해 포장되게 되지만 사과가 아니거나 상태가 좋지 않은 사과라면 기계 이미지 분석을 통해 걸러지게 된다.

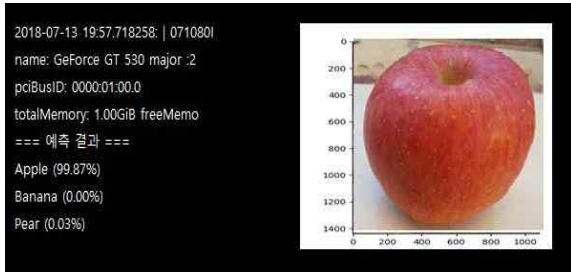


그림 4. 사진예측 결과1
 Figure 4. Photo prediction results1

위 그림처럼 해당 과일이 무슨 과일인지 정확하게 이미 분석을 통해서 알 수 있다. 그리고 위 촬영된 사과가 정상적인 사과가 맞는지 테스트를 통해서 알 수 있다. 만약 그림 정상적인 사과과 아닐 경우 이미지 분석이 어떻게 이루어지는지 살펴 보았다.

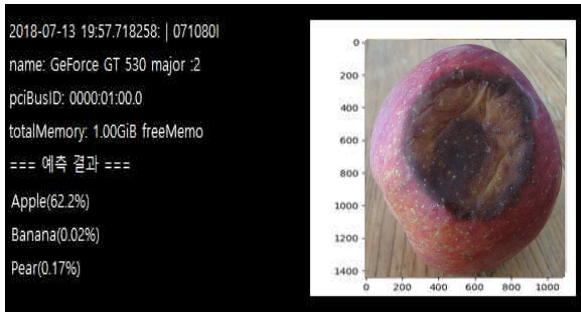


그림 5. 사진예측 결과2
 Figure 5. Photo prediction results2

예측 결과 확인시 사과 대조율이 62.2프로로 사과 판정 기준 80프로를 넘기지 못하기 때문에 기계가 자동적으로 거르게 된다.

표 1. 과일사진 분별 코드
 Table 1. Fruit photo discrimination code

	<pre>import tensorflow as tf import matplotlib.pyplot as plt import matplotlib.image as mpimg import sys tf.app.flags.DEFINE_string("output_graph", "breads_graph.pb", "학습된 신경망이 저장된 위치") tf.app.flags.DEFINE_string("output_labels", "breads_labels.txt", "학습할 레이블 데이터 파일")</pre>
--	---

```
tf.app.flags.DEFINE_boolean("show_image",
    True,
    "이미지 추론 후 이미지를 보여줍니다.")
FLAGS = tf.app.flags.FLAGS

def main():
    labels = [line.rstrip() for line in
tf.gfile.GFile(FLAGS.output_labels)]
    with tf.gfile.FastGFile(FLAGS.output_
graph, 'rb') as fp:
        graph_def = tf.GraphDef()
        graph_def.ParseFromString(fp.read())
        tf.import_graph_def(graph_def,
name='')
    with tf.Session() as sess:
        logits = sess.graph.get_tensor
_by_name('final_result:0')
        image = tf.gfile.FastGFile(
sys.argv[1], 'rb').read()
        prediction = sess.run(logits,
{'DecodeJpeg/contents:0': image})

        print('=== 예측 결과 ===')
        for i in range(len(labels)):
            name = labels[i]
            score = prediction[0][i]
            print('%s (%.2f%%)' % (name,
score * 100))

        if FLAGS.show_image:
            img = mpimg.imread
(sys.argv[1])
            plt.imshow(img)
            plt.show()
if __name__ == "__main__":
    tf.app.run()
```

컴퓨터에게 미리 사전에 교육시킨 데이터에 대해 예측을 시키게 하는 파일으로써, 먼저 flag를 정의하여 데이터를 편리하게 사용할 수 있도록 하였으며, main 함수에서 labels라는 리스트에 실제 학습시켰던 label들의 목록을 가져와 저장시킨다. 이후 파일들을 불러와 모델

을 복원시키며 곧바로 세션을 활성화시켜 예측을 실행한다. 예측이 끝나면 결과값으로 이름과 확률이 나오는데 확률의 경우 소수점 이하의 자리로 나오기 때문에 100을 곱하여 출력시킨다.

3.2 시스템 실행 방법

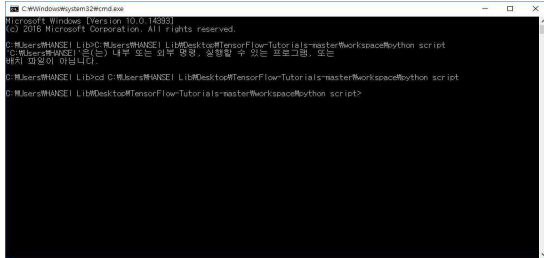


그림 6. 3개월간 사과가격 예측그래프
Figure 6. Three-month apple price forecast graph

터미널 또는 명령 프롬프트를 연 뒤 predict.py 스크립트 파일이 있는 곳으로 이동하여 테스트 할 사진을 workspace/test 경로에 저장하여 명령실행python predict.py ./workspace/test/test1.jpg

표 2. 사과 사진 분석 코드
Table 2. Apple photo analysis code

```
import tensorflow as tf
import os
import glob
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import sys
tf.app.flags.DEFINE_string("output_graph",
                           "./breads_graph.pb",
                           "workspace/breads_graph.pb")
tf.app.flags.DEFINE_string("output_labels",
                           "./breads_labels.txt",
                           "workspace/breads_labels.txt")
tf.app.flags.DEFINE_boolean("show_image",
                             True,
                             "show_image")
FLAGS = tf.app.flags.FLAGS
def main():
    labels = [line.rstrip() for line in
```

```
tf.gfile.GFile(FLAGS.output_labels)]
    with tf.gfile.FastGFile(FLAGS.output
_graph, 'rb') as fp:
        graph_def = tf.GraphDef()
        graph_def.ParseFromString(fp.read())
        tf.import_graph_def(graph_def,
name='')

files=glob.glob(sys.argv[1]+os.path.sep+"*.jp
g")
    with tf.Session() as sess: logits =
sess.graph.get_tensor_by_name('final_result:
0')

    for idx in files:
        if(idx==0):
            print('====Error====')
            break
        image = tf.gfile.FastGFile(idx,
'rb').read()
        prediction = sess.run(logits,
{'DecodeJpeg/contents:0': image})
        print('=== result ===')
        for i in range(len(labels)):
            name = labels[i]
            score = prediction[0][i]
            print('%s (%.2f%%)' %
(name, score * 100))

if __name__ == "__main__":
    tf.app.run()

if FLAGS.show_image:
    img = mpimg.imread
(sys.argv[1])
    plt.imshow(img)
    plt.show()
if __name__ == "__main__":
    tf.app.run()
```

한 장의 사진을 예측할 때와 달라진 것은 과 일을 읽어들이기 때 해당 경로에서 jpg형식의 과

일들을 모두 불러와 파일의 개수를 files라는 변수에 저장시켰으며, 세션이 활성화될 때 for문을 files 값만큼 반복시켜서 파일들을 불러오게 된다.

이후 결과값을 출력할 때도 읽어들이는 파일의 개수만큼 예측 결과값이 출력된다. 단, 파일의 개수가 0이라면 “=====Error=====”이 출력되며 종료된다. 아래의 그림에서 실행 결과를 확인할 수 있다.

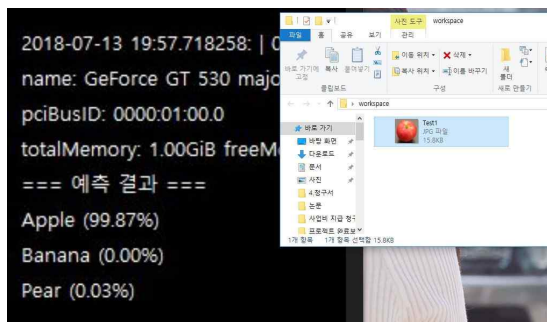


그림 7. 사진 분석 이미지 결과
Figure 7. Photo analysis image results

V. 결 론

최근 Deep Learnin Deep Learning 기술을 활용하여 연구하던 도중 이미지 분석에 관심이 생겨 현 본 프로젝트를 생각하게 되었고, 진행하면서 Inception V3모델은 이미지 분류만 가능하기 때문에 한 장의 사진에서 여러 과일들의 영역을 추출하여 예측시킬 수 없다는 한계에 부딪혀 중간에 darknet 툴을 이용하여 시도해보았으나, 시스템 자원 한계 등의 이유로 좌절할 수밖에 없었다.

하지만 OpenCV 라이브러리를 이용하여 영역을 검출해낼 수 있다는 사실을 알게 되어 컴퓨터가 예측하기 전 먼저 사과외곽선 검출을 통해 JPG 형식의 사진 파일로 저장하여 예측시킴으로써 우리가 부딪혔던 한계를 돌파하려 시도하였으나, RGB값 비교를 통해 사과의 영역을 검출해내는 것을 목표로 진행하였지만, 실제 적용할 분류기가 없어 객체의 최소 크기와 최대 크기를 정하여 영역을 검출할 수 바깥에 없었다.

본 연구의 기대효과로 Deep Learning 기술의 무구한 가능성을 통해 여러 종류의 과일을 컴퓨

터에게 학습시킴으로써 농장주나 작업자가 과일을 수작업으로 직접 분류할 필요성이 없어지며 또한 농장주는 그에 대한 엄청난 인건비 절감으로 인해 품질 좋은 과일들의 저렴한 가격 형성 효과를 기대한다. 나아가 과일의 이미지별 데이터들을 DB에 저장해 추후 원산지나 추가적인 데이터들을 공유함으로써 원산지와 유통과정이 한번에 관리될 수 있는 통합 모델 개발을 목표로 본 연구를 진행하고 있다.

References

- [1] S. W. Kang, S. Y. Lee, J. W. Park “Fire Image Processing Using OpenCV”, The Korea Contents Society, Vol. No.1, pp79-82, 2009
- [2] Y. T. Back, J. S. Kim, K. M. Lee, S. Jin, S. H. Lee, “Intelligent missing persons index system based on the OpenCV and TensorFlow”, Korean society of computer and information, Vol. 24, No. 2, pp59-62 July 2016
- [3] E. J. Jo, J. S. Leem, “Development of Museum Exhibit Recognition Mobile Application using Deep Learning”, The Institute of Electronics and Information Engineers, pp1164-1167 June 2018
- [4] J. W. Kim, “Deep learning algorithms and applications”, Korean Institute of Information Scientists and Engineers, Vol. 33, No.8. pp25-31, August 2015
- [5] Y. J. Song, J. W. Lee, J. W. Lee, “Performance Evaluation of Price-based Input Features in Stock Price Prediction using Tensorflow”, Korea Information Science Society, Vol. 23 No. 11. pp.625-631, 2017
- [6] C. H. Kim, S. W. BYOUN, “Accuracy of the Automating Program of Log Scaling”, Korea Society Of IT Services, Vol.12 , No.4, 2013