

SevenEye 3.0 Server for Linux

사용 설명서 v3.0.5.4(r210929)

문서 Version 관리 표

Release	일자	내 용	전체 Page	비고
r160615	16.06.15	- SevenEye Server for HP-UX	13	
r160819	16.08.19	- 사진 품질검사 라이브러리 추가	22	
r161007	16.10.07	- 사진 품질검사 Score_flag 추가	23	V4
		- 사진 품질검사 함수 변경		
		- 안드로이드 64비트 지원		
		- 안드로이드 라이브러리 파일 통합		
r161207	16.12.07	- 사진 품질검사 안드로이드 jniLib 패치	23	V4
r170602	17.06.02	- 안드로이드 품질검사 사이즈 최적화 - 플랫폼 기준 5MB -> 1MB	24	AOS v2.1.5
		- 매뉴얼 업데이트		
r170609	17.06.09	- 안드로이드 Score_flag 수정	24	AOS v2.1.5.1
		- 아이폰 Score_flag 수정		iOSv2.1.4.1
	17.06.22	- 매뉴얼 업데이트	24	
r170713	17.07.13	- iOS 품질검사 사이즈 최적화	24	V2.1.5.1
r181203	18.12.03	- Server 특징 검출 모델 추가	25	V2.2.5.1
r190131	19.01.31	- 모바일 사진 품질 검사 성능 향상 - Mobile 흑백사진 확인 보안 기능 추가	38	V2.2.5.2
r190628	19.06.28	- 서버 흑백사진 확인 보안 기능 추가	38	V2.2.5.2
r190813	19.08.13	- 모바일 사진 품질 검사 성능 향상	42	V2.2.5.3
r191118	19.11.18	- 사진 품질 검사 기능 개선 - 모바일 품질검사 서버 버전으로 통합	14	V2.2.5.4
r191118	21.01.19	- 경로 직접 설정 방법 추가	16	V2.2.5.4
r210929	21.09.29	- 운전면허증 딥러닝 알고리즘 호환	15	V3.0.5.4
		- 주민등록증 딥러닝 알고리즘 호환		

목차

1. 사용권 계약서.....	4
2. 제품 개요 및 구성.....	5
3. 제품 설명.....	7
4. Server 설명서.....	8

사용권 계약서

소프트웨어 제품 사용권

본 소프트웨어 제품은 저작권 관련법과 국제 저작권 협약을 비롯하여 지적 재산권 법률 및 협약에 의해 보호됩니다.

소프트웨어 제품은 판매되는 것이 아니라 그 사용이 허가되는 것입니다.

1. 사용허가

본 사용권 계약서에 계약서에 동의하는 경우 귀하에게 다음과 같은 권리가 허가됩니다.

한국인식산업(주)는 귀하가 정당하게 취득한 SevenEye 3.0 Enterprise Server 또는 Business Server를 사용할 경우, 별도의 계약서에 명기된 사용자 수에 대하여 본 SevenEye 3.0 Server를 사용할 권리를 허용합니다.

2. 용도제한

본 SevenEye 2.1의 Server 모든 복사본에는 저작권 정보가 표시되어야 합니다.

- 귀하는 본 SevenEye 2.1 Server의 복사본을 제 3자에게 배포해서는 안됩니다.
- 귀하는 관련 법규에서 명시적으로 허용된 제한적인 범위 이외에는 본 SevenEye 3.0 Server를 리버스 엔지니어링, 디컴파일 또는 디어셈블 할 수 없습니다.
- 귀하는 본 SevenEye3.0 Server를 판매, 대여 또는 임대 할 수 없습니다.

3. 계약서의 종료

귀하가 본 소프트웨어 사용권 계약의 내용을 준수하지 않는 경우, 한국인식산업(주)는 여타 권리를 침해하지 않고 동 계약서를 종료 할 수 있습니다.

4. 계약서의 인정

귀하는 소프트웨어 사용권 계약서의 모든 내용을 읽고 이해하였으며, 상기 계약 조건에 동의합니다.

제품 개요

1. Server에서 제공하는 사진이미지를 이용, 신분증 발행기관 사진인식 서버에서 1:1 사진 비교를 하기 위해 특징점을 추출하는 프로그램(Library)으로 추출 된 특징점은 사진인식 서버에서 신분증 원본 특징점과 비교를 하여 신분증 사진의 위.변조 여부를 알려주는 기능을 제공합니다.
2. 본 제품은 비대면 신분증 진위확인 사진인식 서버용 FE 프로그램인 SevenEye Server Library가 제공됩니다.

제품 구성

1. LINUX 라이브러리
2. LINUX JAVA 샘플

제품 설명

3

1. 첨부된 Library는 비대면 신분증 진위확인 사진인식 서비스를 이용을 위한 서버에 설치하여 사용합니다.
2. 본 제품을 이용한 비대면 신분증 진위확인 사진인식 서비스는 첨부 Library를 이용하여 사진의 특징점을 생성한 후, 금융결제원으로 특징점을 전송하면, 금결원과 연계된 신분증 발행기관 사진인식 서버에서, 각 금융 기관에서 올라온 특징점과 사진인식 서버에서 생성된 원본 사진의 특징점을 비교하여 일치/불일치 여부를 알려주는 방식으로 이루어집니다.
3. 최종 고객에게 제공되는 Library는 해당 금융 기관 전용으로 Build된 Library로, 반드시 최종 고객과 동일한 전용 Library를 사용하여야 합니다.
4. 사용순서
 - 1) 스마트 폰 어플리케이션 또는 서버에서 신분증의 사진을 Crop 합니다.
 - 2) Crop 사진을 금융기관용 BMP로 저장
 - ✓ BMP 권고 해상도 및 규격 : 380*460, 24Bit BMP 523KB
 - 3) 서버에서 품질검사 Library를 사용 사진 품질검사 수행 (Mono Image)
 - 4) 품질검사에서 이상이 없는 경우 사용 BMP파일에 대한 특징점 생성
 - 5) 특징점 데이터를 BASE64로 변환
 - 6) BASE64로 변환된 특징점을 금융결제원으로 전송
 - 7) 신분증 발급 기관 서버에서 원본 특징점과 비교
 - 8) 사진 비교 결과를 금융기관에서 수신
6. 사진 특징점 추출과 품질검사는 각 각 1초 이내에 결과 값을 제공합니다.

서버 설명서

◆ 사진특징점 추출 LINUX 라이브러리 설명서

1. 디렉토리 설명

KorealD /

Lib64 : 라이브러리 폴더

Bio : 자바 클래스 폴더

Data : 얼굴검출용 필터용 기본 데이터

model: 얼굴검출용 필터용 기본 데이터

Log: 로그 저장 폴더

Log/Image2 : 입력 이미지 저장 폴더

Face_Demo.java : 자바 샘플 프로그램

- 대,소문자 주의

2. 설치과정

- 1) 라이브러리 포함된 KorealD 폴더를 설치 대상 서버에 복사한다.

예) **home/dev/** 폴더에 복사하는 경우

/home/dev/KorealD

/Data

/java

/libup64h

/Log

/model

- 2) 자바 라이브러리 경로 설정 (so 가 있는 경로를 설정 함)

export LD_LIBRARY_PATH=/ /home/dev/KorealD/libup64:\$ LD_LIBRARY_PATH

- 3) Log 경로 설정 (선택 사항 : 유지보수 용)

/home/dev/KorealD/Data/savepath 파일을 열어서 로그파일이 쌓이는 경로를 설정한다.

경로 설정이 없는 경우 로그가 쌓이지 않음

로그 설정을 하면 주기적으로 삭제를 해야 함

Vi savepath

/home/dev/KorealD/Log

서버 설명서

◆ 사진특징점 추출 LINUX 라이브러리 설명서

4) java 소스에서 경로를 수정한다.

```
cd /home/dev/KorealD/java
vi face_Demo.java
    face_Demo.java 를 열어서
// Data 폴더가 있는 경로를 입력한다.
    String confPath = "/home/dev/KorealD/";
// cmmSVR_FA.so 파일이 있는 경로를 입력한다.
    String cmmLibPath = "/home/dev/KorealD/libup64h/";
```

5) 실행

```
[root@kf3 facebio]# javac ./bio/face/Faceprint.java
[root@kf3 facebio]# javac face_Demo.java
[root@kf3 facebio]# java -Djava.library.path=/home/devibs1/KorealD/libup64h -d64
face_Demo 1 dbimage Size97531
/home/devibs1/KorealD/libup64h
program start
kf3_init_mutex start
Load_Lib
kf3 model path = /home/devibs1/KorealD//model License check OK
KII_Init_det _ENCORD_SEVENEYE start
detect model rt1=0
detect model rt2=1735840
KII_Init_det _ENCORD_SEVENEYE end
KII_Init_det rt=0
KII_Init KII_Init_det rt=0
feature model rt1=0
feature model rt2=4091592
KII_Init_fet rt=0
KII_Init KII_Init_fet rt=0
KII_Init OK
kf3_init_mutex end
ExtractDB return=89
program end
KII_End()
SevenEye_Core_Uninit()
```

(특징점 점수는 89점 , 특징점 파일 1_NonFace_feature.f 가 생성됨 , 생성된 파일 사이즈는 3500 바이트)

서버 설명서

◆ 사진특징점 추출 LINUX 라이브러리 설명서

6) 프로그램 실행 내용 로그 확인

cd /home/dev/KorealD/Log

```
15:36:43.163 | VER [3.0.5.4_3] 201512243
15:36:43.163 | start cmmFA_ExtractDB id=1234 kf3path=/home/devibs1/KorealD/libup64h/
path=/home/devibs1/KorealD/
15:36:43.177 | Core initialization ok
15:36:43.212 | kf3 initialization [ OK ]
15:36:43.212 | input image size = 97531
15:36:43.212 | K_Extract start
15:36:43.249 | K_Extract End return=0
15:36:43.253 | cmmFA_ExtractDB,CreateRawImageFromJPG
[ true ] ,RawWidth=354,RawHeight=433
15:36:43.253 | Image to raw OK
15:36:43.256 | Mono Check return MonoPercent=36
15:36:43.256 | Mono Check : is COLOR
15:36:43.279 | face detect [ OK ]
15:36:43.279 | SevenEye FacialImage QA=89
15:36:43.279 | EYE 140,187,216,190
15:36:43.336 | SevenEye_FacelImage2Template bResult=1
15:36:43.336 | Face to Template ok
15:36:43.466 | Eye_Find is 139 189 215 191
15:36:43.466 | get face qa 2 result = 89
15:36:43.466 | get face qa 3 result = 89
15:36:43.466 | add kf3 feature 516 byte
15:36:43.468 | Detect End
15:36:43.468 | End return=89

(15:36:43.473),(3.1.5),##### program end #####
(15:36:43.474),(3.1.5),##### KII_End() #####
(15:36:43.474),(3.1.5),##### SevenEye_Core_Uninit() #####
```

서버 설명서

◆ 사진특징점 추출 LINUX 라이브러리 설명서

3. 파일 설명

1) [so 파일] - 반드시 필요한 파일

cmmSVR_FA.so :c++ 동적 라이브러리.

libFaceprint.so : java 에서 동작되는 라이브러리 파일 , 라이브러리 PATH 폴더에 복사해야 합니다.

(제우스 WAS 사용시 libFaceprint.so 파일을 제우스 시스템 폴더에 복사하면 됩니다)

2) [Data폴더] 및 [model]

Data 및 model 폴더 이름을 변경하면 안됨

* 얼굴 검출용 필터파일을 엔진에서 불러와야 하므로 필터 데이터가 있는 절대 경로를 지정해줘야 합니다.

* 지정방법은 자바에서 얼굴 검출 함수(jniFAExtractDB)호출시 첫번째 파라메타에 경로를 넣으시면 됩니다.

예) Data 및 model 폴더가 있는 상위 폴더를 입력한다(마지막 '/' 입력)

```
fp.jniFAExtractDB("/KorealD/" ,.....
```

지정된 경로 밑에는 Data 폴더가 있고 Data 폴더 밑에는 필터용 파일들이 있어야 한다.

경로가 입력 되지 않으면 에러로 리턴 됨 (-4)

경로에 Data 폴더가 없고 필터용 파일이 없으면 에러로 리턴 됨 (-2)

3) [로그 파일]

log경로는 Data 폴더의 savepath 파일에 지정하면 된다.

로그를 쌓지 않으려면

Data 폴더의 save 파일의 첫번째 값을 'Y'에서 'N' 으로 변경하면 된다.

특징점 결과가 0 이거나 -3 이 나오면 입력 이미지 에러 이므로 입력이미지를 로그에 저장한다.

이미지 로그 경로는 log 경로 아래에 Image2 폴더에 일자별로 폴더가 생성된다.

이미지 로그를 남기지 않으려면 Data 폴더의 save 파일의 두번째 값을 'Y'에서 'N' 으로 변경하면 된다.

4. [코드값]

1) [에러 리턴값]

-2: 엔진 초기화 오류

-3: 이미지 처리 오류

-4: 환경설정폴더 경로 오류

-5: 메모리 할당 오류

-6: 라이선스 오류

-88: 눈검출 엔진 초기화 오류

-89: 사진이 칼라가 아님

-105 : 연동 키값 오류

-106 : data 경로 오류

-100 : cmmSVR_FA.so 라이브러리 로딩 오류

-200 :cmmFA_ExtractDB 함수 로딩 오류

-401 : 이미지 크기 오류

-501 :눈검출 오류

서버 설명서

◆ 사진특징점 추출 LINUX 라이브러리 설명서

5. [함수 설명]

1) `int jniFAStart(String cmmLibPath);`

라이브러리 초기화 함수, 프로그램 시작 시 한번 실행

cmmLibPath 는 cmmSVR_FA.so 파일 위치 경로 지정, 마지막 '/' 입력,
예)"/home/devibs1/KorealD/lib64/"

리턴 : 0(정상) , -100(라이브러리 cmmSVR_FA.so 로드 실패), -200(함수 로드 실패)

2) `int jniFAEnd();` 라이브러리 메모리 해제 함수, 프로그램 종료 시 한번 실행한다.

리턴 : 0(정상) , 나머지는 실패

```
int jniFAExtractDB(String confPath,  
                   String unicode,  
                   byte[] Image,  
                   int nImageSize,  
                   int nFormat,  
                   byte[] DBFeature);
```

3) 특징점 추출하는 함수

입력 :

가) `confPath`: Data 폴더와 model 폴더가 있는 경로 + "/"

나) `unicode` : 로그 저장용 트랜잭션 번호

다) `nImageSize` : 이미지 크기

라) `nFormat` : 이미지 타입 , Jpg:1 , bmp:2 를 입력

마) `DBFeature` : 특징점을 받을 메모리 버퍼 변수 입력, 3500 바이트 , 바이너리로 출력됨
이것을 베이스 64로 변환해서 사용해야함

리턴 : 특징점 점수 (1 ~ 100),

실패시는 0 또는 0 보다 작은 값이 출력됨

1 ~ 100까지의 점수는 내부 관리를 목적으로 사용하는 점수입니다.

점수가 높을수록 사진 품질이 양호하여 사진인식 성공률이 높아지나, 특정한 점수 지표는 사용하지 않습니다.

4) `jniFAEnd();`

라이브러리 메모리 해제 함수

종료 시 한번 사용

서버 설명서

◆ 사진특징점 추출 LINUX 라이브러리 설명서

6. [소스]

[bio/face/Faceprint.java]

```
package bio.face;
```

```
public class Faceprint {
```

```
    public native int jniFAStart(String cmmLibPath);
```

```
    public native int jniFAEnd();
```

```
    public native int jniFAExtractDB(String confPath,
```

```
String unicode,
```

```
byte[] Image,
```

```
int nImageSize,
```

```
int nFormat,
```

```
byte[] DBFeature);
```

```
    static {
```

```
        System.loadLibrary("Faceprint");
```

```
    }
```

```
}
```

[face_Demo.java]

```
import java.io.*;
```

```
import java.util.*;
```

```
import java.nio.ByteBuffer;
```

```
import bio.face.Faceprint; // 패키지명은 변경하면 안됩니다.
```

```
/////
```

```
class face_Demo {
```

```
    public static void main(String[] args) throws IOException{
```

```
        Faceprint fp;
```

```
        fp = new Faceprint();
```

```
String cmmLibPath = "/KorealD/lib64/";
```

```
fp.jniFAStart cmmLibPath ); // 프로그램에서처음한번만한다.
```

서버 설명서

◆ 사진특징점 추출 LINUX 라이브러리 설명서

6. [소스]

```
String dbfilename = "1.bmp";
String confPath = "/KorealD/";
String unicode = "10000001";
byte[] DBFeature = new byte[3500]; // db 특징점
int len, dlen;
int status;

// db 이미지파일에서읽어오기
FileInputStream fin = new FileInputStream(dbfilename);
int count = fin.available();
byte[] dbimage = new byte[count];

dlen = fin.read(dbimage);
String str = "dbimage 크기" + dlen ;
System.out.println(str);

System.out.println(System.getProperty("java.library.path"));

if (DBFeature != null )
{
    // 특징점추출
    int rt2 = fpjniFAExtractDB(confPath, unicode, dbimage, dlen, 2, DBFeature);

    str = "ExtractDB return=" + rt2 ;

    System.out.println(str);

    // 특징점저장
    FileOutputStream out1 = new FileOutputStream("1_db_feature.bin");

    out1.write(DBFeature);
    out1.close();
}

fpjniFAEnd(); // 프로그램 종료 시 한번 만 수행한다.
}
```



한국인식산업[주]

Korea Identification Inc.

기술문의

한국인식산업(주) 기술연구소 전용하 실장

02-581-3511

010-8332-9074

yhjun@korea-id.co.kr