

박사학위논문

비문 활용을 위한 BEBLID Descriptor 값
도출 및 객체인지 매칭 기술 고도화 연구

2024년

한 성 대 학 교 대 학 원

스마트융합컨설팅학과

스마트융합컨설팅전공

양 전 성

박 사 학 위 논 문
지도교수 이석기

비문 활용을 위한 BEBLID Descriptor 값 도출 및 객체인지 매칭 기술 고도화 연구

Research on Derivation of Optimal BEBLID Descriptor
Value for Muzzle Pattern Utilization and Advancement of
Object Recognition Matching Technology

2024년 6월 일

한 성 대 학 교 대 학 원

스마트융합컨설팅학과

스마트융합컨설팅전공

양 전 성

박 사 학 위 논 문
지도교수 이석기

비문 활용을 위한 BEBLID Descriptor 값 도출 및 객체인지 매칭 기술 고도화 연구

Research on Derivation of Optimal BEBLID Descriptor
Value for Muzzle Pattern Utilization and Advancement of
Object Recognition Matching Technology

위 논문을 컨설팅학 박사학위 논문으로 제출함

2024년 6월 일

한 성 대 학 교 대 학 원

스마트융합컨설팅학과

스마트융합컨설팅전공

양 전 성

양전성의 컨설팅학 박사학위 논문을 인준함

2024년 6월 일

심사위원장 박 현 성 (인)

심 사 위 원 박 상 선 (인)

심 사 위 원 천 성 용 (인)

심 사 위 원 이 상 복 (인)

심 사 위 원 이 석 기 (인)

국 문 초 록

비문 활용을 위한 BEBLID Descriptor 값 도출 및 객체인지 매칭 기술 고도화 연구

한 성 대 학 교 대 학 원
스 마 트 융 합 컨 설 팅 학 과
스 마 트 융 합 컨 설 팅 전 공
양 전 성

세계적으로 반려동물과 관련된 다양한 산업의 등장과 관련 시장이 급성장하고 있으며, 우리나라에서도 1인 가구의 확대와 저출산 및 고령화 등으로 인해 반려동물을 키우는 가구가 지속적으로 증가하고 있다.

이에 따라 반려동물을 체계적으로 보호하고 관리하기 위해 동물보호법에 따른 동물등록제도를 2014년부터 시행하고 있으며, 대표적으로 내·외부 인식장치 및 등록태그를 부착하여 등록을 진행하고 있으나 등록의 불편함과 IC 칩을 피하부위의 삽입하여 내장에 따른 비윤리성으로 인해 반려동물의 등록률은 등록률에 대비하여 약 37% 정도로 미미한 실정이다.

본 연구에서는 기존 등록 방식의 불편함을 해소하기 위해 사람의 지문과 같이 독특하게 나타나는 개의 비문(鼻紋, Muzzle Pattern)을 기반으로 등록률을 제고하는 기술에 대해 연구하였다. 비문 데이터의 수집, 저장 및 분석을 통해 비문 인식 기술의 매칭률 정확도를 향상시킬 수 있는 방

안을 제시하는 것을 목표로 하였으며 이미지의 특정 영역에 대한 객체 인식 알고리즘인 SIFT, BRIEF, ORB, BRISK을 비교 분석하여 대표적인 ORB를 기반으로 BEBLID Descriptor를 적용하는 방식을 연구하였다.

제안된 방법을 입증하기 위해 분석을 위한 자료로 총 200개의 비문 데이터 셋을 구축하였으며 효율적인 객체인식을 위해 기존 ORB 알고리즘에 BEBLID의 정합을 통해 정확도 향상과 매칭시간의 절감을 유도하였으며 찾아낸 특징 수를 복합적으로 고려한 새로운 매칭 기준을 제시하고, 이를 기준으로 서로 다른 알고리즘의 조합이 얼마나 효율적인지를 탐구했다.

분석의 결과는 BEBLID Descriptor 값을 가장 기본값인 1로 설정 하였을 경우에 제안된 방법론의 매칭점 인식률은 76.24%로 나타났고, 비교 대상인 기본 ORB 알고리즘은 66.82%인 것으로 나타났다. 또한 본 연구에서는 실험 데이터 셋에 대한 최적의 BEBLID Descriptor 값을 찾아내기 위해 그 값을 0.25~2까지 가변적으로 조정하면서 실험을 하였고 그 결과로 최적의 BEBLID Descriptor값은 0.75로 나타났으며 이 때 BEBLID의 매칭점 인식률은 87%로 타 방식에 비해 높게 나타났다.

결과적으로 데이터셋에 대하여 최적의 BEBLID Descriptor값을 도출 하여 적용할 경우 기존 ORB 기법에 비해 인식률에서 큰 향상이 있다는 것을 입증하였으며 본 연구에서 제안하는 비문 인지율 향상 기술을 실제 상용화하여 반려견 등록 및 관리에 사용할 경우, 현재 활성화가 시급함에도 저조한 사용 현황으로 문제점을 안고 있는 동물 등록제도의 등록률을 향상시키고 효율적인 등록관리가 가능할 것이다.

【주제어】 : BEBLID 알고리즘, Descriptor, 바이오 인식, 비문 패턴, 특징점 추출

목 차

I. 서 론	1
1.1 연구의 배경 및 목적	1
1.2 연구의 구성	6
II. 이론적 배경	9
2.1 객체 인지 기반 기술	9
2.2 객체 인식 알고리즘	15
2.3 비문 기반 객체 인지 기술	27
2.4 연구의 차별성	35
III. 연구 방법	37
3.1 분석 방법	37
3.2 연구 주요 내용	39
3.3 자료수집 및 연구 분석	40
3.4 비문 분석 프로세스	43
IV. 연구결과	57
4.1 비문 표본의 특성 분석	57
4.2 실험 결과	61
4.3 적용 프로그램 함수 분석	68
4.4 EVELID 알고리즘의 적정 Descriptor 값 산정	73
4.5 데이터 매칭률 조사	80
4.6 비문 데이터 매칭 결과	88
V. 결 론	93
5.1 연구의 결론	93

5.2 연구의 한계점 및 향후 연구방향	96
참고문헌	99
ABSTRACT	105

표 목 차

[표 2-1] 대표적 비문 연구 내역	34
[표 3-1] DB 데이터 샘플 정보(10개 데이터셋)	42
[표 3-2] 입력 데이터 테이블 구조	45
[표 3-3] 출력 데이터 테이블 구조	46
[표 4-1] 해상도 기준 데이터 분류	59
[표 4-2] 확정 시험 데이터 분류	60
[표 4-3] 비문 데이터 매칭 시간	67
[표 4-4] ORB와 Descriptor 함수 비교	71
[표 4-5] 알고리즘 기반 BEBLID 값 적용 실증	74
[표 4-6] 가변 Descriptor 값에 따른 알고리즘별 매칭률 비교	76
[표 4-7] 동일 비문 매칭 결과	82
[표 4-8] 동일 비문 이미지 매칭 비율	83
[표 4-9] 비동일 비문 매칭 결과	85
[표 4-10] 비동일 비문 이미지 매칭 비율	86
[표 4-11] 데이터 매칭 그래프	89

그 립 목 차

[그림 1-1] 비문데이터 세부 모습	5
[그림 1-2] 연구 과정	8
[그림 2-1] 객체 인식 딥러닝 절차	11
[그림 2-2] 소프트맥스 손실함수	12
[그림 2-3] 거리기반 손실함수	13
[그림 2-4] 앵귤러 마진 기반 손실 함수	14
[그림 2-5] SIFT 알고리즘	15
[그림 2-6] SURF 알고리즘 구조	18
[그림 2-7] BRIEF 알고리즘	20
[그림 2-8] FAST 특징 추출 알고리즘	22
[그림 2-9] 생체 인식 서버 모델	29
[그림 2-10] 생체 인식 클라이언트 모델	29
[그림 3-1] 비문 매칭 프로세스	38
[그림 3-2] 비문 매칭 시험 환경	40
[그림 3-3] 비문의 특징점 바이너리 정보 저장	41
[그림 3-4] 비문 분석 전체 프로세스	43
[그림 3-5] 비문 ROI 설정	44
[그림 3-6] Binary description 추출	45
[그림 3-7] 특징 저장 데이터 셋	46
[그림 3-8] 비문 데이터 교정	47
[그림 3-9] 이미지 히스토그램(a) light Reflection (b) Dark Reflection	47
[그림 3-10] 비문 이미지 변환	50
[그림 3-11] 블록 단위의 히스토그램 균일화(CLAHE)	51
[그림 3-12] CLAHE 처리 비문 이미지 특징점 추출	52
[그림 3-13] 비문 특징점 매칭(knn Match)	52

[그림 3-14] 해밍 거리 정의	53
[그림 3-15] 정규화된 히스토그램(확률 분포)	54
[그림 3-16] RANSAC 알고리즘 적용 결과	55
[그림 4-1] 등록된 반려견 데이터 조회 화면	58
[그림 4-2] 비문 매칭 흐름도	61
[그림 4-3] 비문 데이터 등록 정보	62
[그림 4-4] 비문 데이터 검색	63
[그림 4-5] 비문 매칭 결과	64
[그림 4-6] 인터페이스 연계 시간	66
[그림 4-7] API 기반 데이터셋 매칭 확인	69
[그림 4-8] ORB 사용 정상 매칭률	70
[그림 4-9] BEBLID 사용 정상 매칭률	71
[그림 4-10] Data API 기반 데이터 서치	72
[그림 4-11] 비문 매칭 데이터 비교	73
[그림 4-12] Descriptor별 매칭률 그래프	77
[그림 4-13] 알고리즘별 매칭률 그래프	77
[그림 4-14] BEBLID 기반 이미지 매칭 실험	78
[그림 4-15] Keypoint 이미지 판별	79
[그림 4-16] 비문의 특징점들 비교 분석 프로그램	80
[그림 4-17] 특징 디스크립터 매칭 프로그램	81
[그림 4-18] 비문 데이터 간 매칭 흐름도	88

I. 서론

1.1 연구의 배경 및 목적

1.1.1 연구의 배경

최근 들어 세계적으로 반려동물 관련 산업과 시장이 빠르게 성장하고 있으며, 국내에서는 1인 가구의 확대와 저출산 고령화 등의 영향으로 반려동물을 키우는 가구가 지속적으로 증가하고 있다. 국내 반려동물 보유 가구는 2023년 1인 가구 중 9.7%에 해당하는 552만 가구, 인구수 1262만 명으로 국내 4인 가구 중 1가구가 반려동물을 기르며 매년 증가 추세이다(황원경 & 이신애, 2023).

이에 반려동물의 체계적 관리를 위해 2014년도부터 동물보호법에 따른 동물등록제를 시행하고 있다. 대표적으로 체내 무선인식 기기 삽입과 등록 태그를 부착하는 방식이 있지만 실제 동물등록제 등록률은 2018년 20.5%, 2019년 24.4%, 2020년 27%, 2021년 37.4%로 미진한 상황이다(농림축산식품부 동물등록률 자료, 2022).

반려동물의 등록 방식 중 내장형 무선인식 방식은 몸 안에 칩을 넣는 방식이며 외장형 및 등록 인식표 부착 방식은 동물 개체정보가 담긴 목걸이 또는 인식표를 부착하는 방식이다. 칩 방식은 반려동물 몸에 무선기능을 탑재한 칩을 내장하는 방식으로 통증 유발과 반인륜적 생각으로 사용을 기피하며, 목걸이 인식표는 분실의 위험과 외출 시 부착에 대한 번거로움 때문에 이용성이 낮게 나타난 것으로 분석되었다. 이러한 불편함을 해소하고 반려동물의 체계적인 등록률을 제고하기 위하여 Biometrics 기법을 활용한 방식이 대안으로 등장하였으며 이는 반려동물이 가진 유일한 인식키인 동물의 작은 코주름, 즉 비문(鼻紋, Muzzle pattern)를 활용하여 특징

점을 추출하여 객체의 식별 또는 인증을 하는 기술이다.

지난 수십 년간 객체 인식 기술 연구가 활발히 진행되었으며 이는 이미지 속 내부 데이터베이스(DB)에 저장된 객체 중 가장 유사한 분야를 식별 및 매칭하여 인식하는 기술이 핵심이다. 인식 기술은 정규화된 이미지의 특징점을 기반으로 구별하는 방식으로 주로 적용되는 인식 모델은 이미지의 특징점 간의 픽셀의 기울기 벡터, 색차 등 유사성을 비교하여 식별하는 방식을 사용하였다.(김용현, 박원표, 노명철, 신종주, 2020) 인식 기술은 주로 인물, 객체의 고유한 물리적, 행동적 생체정보의 특징 확인을 위해 사용되고 있으며 인식 모델의 추출 단계에서 객체 위치가 다르거나 그 촬영 각도가 다르면 인식 정확도가 낮아지는 문제점을 가지고 있고 특히 소형 객체에는 인식률이 낮은 단점을 가지고 있다.

본 연구에서는 생체 인식기술을 동물의 비문에 적용하여 동물 등록률의 제고시키는 방법을 살펴보고자 하며 기존 외부 무선 주파수 식별 장치와 반려동물 태그 등에 방식의 불편함을 해소 할 것으로 최종적으로 기존등록 방식을 개선하여 등록률을 제고할 것이다.

1.1.2 연구의 목적

본 연구의 목적은 반려견의 축적된 비문 이미지에 대해 ORB 알고리즘과 BEBLID의 Descriptor 값을 가변적으로 적용하여 최적의 인식률을 나타내는 변수값을 도출하는 것이다. 연구를 통해 기존 객체인식 알고리즘의 이미지 매칭에 대한 정확도를 높이고 실행 시간을 줄이는 결과에 대해 실험을 통해 증명하고자 한다.

이의 입증을 위해 본 연구에서는 반려동물의 비문 이미지를 촬영하여 데이터를 200개 이상 수집하며 이를 ORB(Oriented FAST and Rotated BRIEF)와 BEBLID(Boosted Efficient Binary Local Image Descriptor) 알

고리즘을 활용하여 인식률을 향상하는 변수의 최적값을 도출하고자 한다. 기존의 알고리즘이 특징점의 수가 제한적이거나 작은 비문의 매칭에 한계를 나타내었다면 본 연구에서는 2개의 객체 인식 알고리즘 접목을 통해 비문의 전역적 특징을 추출한 후 매칭하는 방식으로 지역적 비교로 인한 특징점의 오인식을 개선하는 효과를 증명하고자 한다.

실험 절차는 총 10단계로 비문 데이터의 수집 후 데이터를 정제하는 과정을 통해 데이터베이스(DB)를 구성하고 특징점을 추출하여 비문 데이터의 인식률을 매칭 후 결과를 도출하는 과정으로 진행한다.

매칭을 위해 ORB에서 해밍 거리(hamming distance)를 사용하여 특징점 간의 거리를 기반으로 부정확성을 줄이기 위해 형태, 크기 등 국부적인 특징점을 추출하여 영상인식에 활용토록 한다. 영상의 특정 구역에 대한 객체 인식 및 매칭 정확도에 대한 검증은 BEBLID의 Descriptor의 가변값에 대해 기존 알고리즘인 SIFT, SURF, BRIEF, BRISK 및 ORB에 각각의 성능지표를 계산하여 인식률을 비교하는 방식을 활용한다.

1.1.2 연구의 방법

본 연구에서 비문 매칭을 위해 특징점 검출(Detection), 특징점 서술(Description), 그리고 특징점 정합(Matching)의 3단계의 과정을 수행하며 비문 데이터를 수집하여 오류를 제거하는 전처리 과정과 특징 추출, 매칭, 후처리 과정을 수행한다. 전처리를 위해 수집된 데이터의 제안된 크기 조정 및 히스토그램 평준화를 사용하며 특징 추출 알고리즘의 경우 SIFT, SURF, BRISK, Oriented FAST 및 ORB가 사용되고 있어 이를 비교 최적의 방식을 선택한다.

이미지의 특정 영역에 대한 객체 인식 기술로는 SIFT (Scale Invariant Feature Transform), SURF(Speeded Up Robust Features), BRISK

(Binary Robust Invariant Scaling Keypoints), Oriented FAST 및 ORB 등 기술이 활용되고 있다. 특징점을 기반으로 하는 객체 인식 영상 매칭 알고리즘에 대한 많은 연구가 이루어졌으며 대표적으로 빠른 객체 인식을 위해 SIFT 알고리즘과 SURF 알고리즘의 문제점을 해결한 ORB 알고리즘이 대표적으로 사용되고 있으며 우수한 인식률을 나타내고 있다.

ORB 알고리즘은 OpenCV Labs에서 개발됐으며 FAST Corner Detection 알고리즘을 사용하여 특징점 추출을 하며 Descriptor 생성에는 Rotation BRIEF 알고리즘을 사용하여 보다 정확한 특징점을 빠르게 추출하여 매칭의 정확도를 향상시켰다. 특징점을 추출하기 위해 Harris Corner Detection, 방향을 결정하기 위한 Intensity Centroid, Descriptor 생성 단계의 BRIEF 알고리즘을 사용, Matching 단계에서는 Hamming distance 알고리즘을 사용한다(Pisano, E. et al, 1998). 비문의 특징점 검출 방법은 특징점을 둘러싸는 일정 크기의 작은 영상 패치 내의 영상 데이터로부터 정보를 추출하여 영상 특징 벡터를 생성하고 해당 특징점의 메타 데이터로 보관하여 특징점들 사이의 유사도 매칭을 통해 구별하도록 한다.

이러한 비문을 이용한 연구로 1926년 J. G. Horning 외 3인이 쓴 Nose Printing Your Cat and Dog Patients가 최초이며 비문에 잉크를 묻혀 종이에 찍어 비교하는 방식에서 출발하여 2018년 Kumar의 DBN(Deep Belief Network)을 활용한 기술까지 발전하였으며 딥러닝을 사용하는 방식을 적용하여 정확도와 매칭률을 향상하였다(Kumar & Santosh, 2018).

본 연구에서는 기존의 특징 추출 알고리즘을 기반으로 최적의 비문 데이터의 추출 및 매칭을 위하여 2020년 옥스포드 대학의 연구 논문에서 소개된 BEBLID 알고리즘의 Descriptor값을 비문과 같이 섬세하고 다양한 형태에 접목하여 반려동물의 매칭률 향상 방안을 연구하고자 한다.

1.1.3 연구의 범위

본 연구에서는 반려동물의 비문 이미지를 촬영하여 데이터를 200여셋 이상 수집하며 이를 ORB와 BEBLID 알고리즘을 활용하여 비교함으로써 인식률을 향상하는 변수위 최적값을 도출하고자 한다. 기존의 알고리즘은 특징점의 수가 제한적이거나 작은 비문의 매칭에 한계를 나타내었다면 본 연구에서는 2개의 객체인식 알고리즘 접목을 통해 비문에 포함된 전역적 특징을 추출한 후 매칭하는 방식으로 특징점의 오인식을 개선하는 효과를 확인하고자 한다.

본 연구에서는 반려견의 비문 이미지를 촬영하여 ORB 알고리즘을 활용한 BEBLID Descriptor의 최적값을 도출하여 개체인식 시스템의 매칭률이 향상되는 고도화 방안에 대하여 실험하고자 한다.

본 연구에서는 각 기술에 대해 분석하여 최적의 매칭 기술로 대두된 BEBLID 알고리즘 기술자를 고도화하여 비문에 적용하는 연구를 진행토록 한다. 기존의 객체 인식 알고리즘이 지정된 구역의 비교를 기반으로 특징점의 일치 여부를 평가하였다면, 본 연구에서는 ORB 알고리즘과 BEBLID Descriptor를 활용하여 비문의 전역적 특징을 추출한 후 저장된 DB의 데이터와 비교하여 매칭률에 대한 고도화 여부를 평가하고자 한다.

[그림 1-1]에서 관찰할 수 있듯이 비문 이미지는 사람의 지문과 같이 독특한 구조의 형태로 땀구멍 및 주름으로 구성되어 있으며 비문의 유일성 근거자료는 많은 연구를 통해 증명되었다.



[그림 1-1] 비문 데이터 세부 모습

대표적으로 2021년 서울대학교와 건국대학교 연구진이 합동으로 연구한 “The Formation and Invariance of Canine Nose Pattern of Beagle Dogs from Early Puppy to Young Adult Periods Animals”에서 10마리의 반려동물을 대상으로 2개월령부터 11개월령까지 비문의 유일성과 영속성을 조사한 결과 비문은 변하지 않고 유일한 성격을 가지며 그 특징이 유효하다는 결론을 제시하였다(최형인, 이운석, 신현진, 이성진, 2021). 따라서 비문은 사람의 지문과 같은 특징을 지니고 있어 반려동물의 객체 인식에는 문제가 없으며 특히 나이가 변화하여 성장하여도 고유의 비문 특징은 변하지 않는다는 것을 증명하였다.

본 연구에서는 수집된 비문 데이터를 특정 구역의 ROI 영역으로 설정하여 특징점을 추출하여 이진화된 모델로 변경한 후 생성된 변환 행렬(Transformation matrix)을 적용하여 상호 비교하였다. 이때 BEBLID의 기술자(Descriptor)의 값을 가변적으로 적용하여 최적의 인식율을 나타내는 최적값을 산정하였다. BEBLID는 이진(Binary)형태의 특징점 기술자를 가지고 있으며 이 기술자는 비문과 같은 작은 이미지에서 실시간으로 특징점을 효율적으로 추출하는 데 사용된다.

본 연구에서는 비문 인식률 향상을 위해 BEBLID 알고리즘의 최적 기술자를 도출하며 이를 통해 비문 인식률의 향상과 반려동물의 등록률을 높이는 방안을 제시하고자 한다.

1.2 연구의 구성

전 세계적으로 1인 가구 확대 및 저출산 고령화 등에 따라 반려동물 보유 가구가 지속적으로 증가하고 있다. 이를 제도적으로 관리하기 위해 동물보호법에 의한 동물 등록제를 실시하고 있으며 주요 방식으로 내·외장형 무선 인식장치, 목걸이 타입 등록 인식표 부착 방식이 있으나 전체 등

록 대상 대비 반려동물에 대한 등록률은 1/5 수준으로 미비한 형편이다. (황원경 & 이신애, 2023).

이는 칩 삽입의 비윤리적 문제 및 목걸이 방식의 탈부착 불편함으로 저조한 등록률을 나타내며 이를 해소하고자 새로운 등록 기술인 바이오인식 기술이 등장하였다. 이는 사람의 지문과 같이 고유하게 나타나는 비문을 인식하는 기술로 카메라를 통해 반려동물의 비문 이미지를 획득하고, 영상 처리를 통해 이미지상의 특징점을 추출하여 추출된 특징점을 기반으로 비문 데이터와 일치 여부를 판단함으로써 동일 여부를 판별하게 하는 기술로 다양한 사업군 창출과 변화를 가져 올 것으로 예상된다.

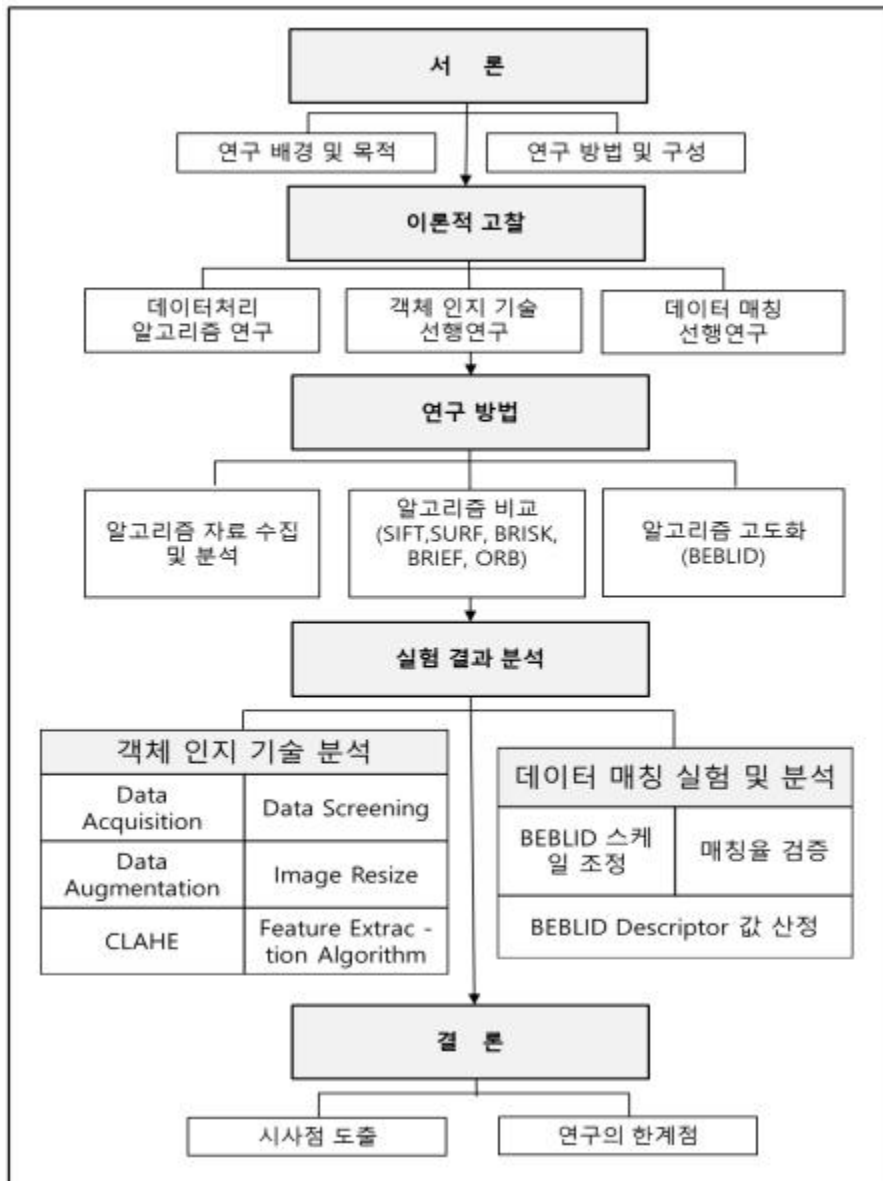
본 연구에서는 객체인지의 다양한 알고리즘을 비교 분석하여 최적의 알고리즘을 반영하며 인식률 제고 알고리즘인 BEBLID를 반려견의 비문인지에 접목함으로써 매칭률 제고 및 관련 기반 기술의 연구를 진행하고자 한다.

이는 비문의 인지율을 향상하는 기술구현을 통해 반려견의 체계적인 관리와 유기로 인한 사회 문제 해소 방안을 제시하며 다양한 데이터를 활용하여 AI·빅데이터 산업에서의 데이터 허브·유통·활용에 이용 가능한 양질의 데이터를 제공함으로써 연구 성과 확대에 연결될 수 있을 것이다.

본 논문의 구성은 서론, 이론적 고찰, 연구 방법 및 분석 결과, 결론 부분의 총 5단계로 구성되어 있다.

1장 서론에서는 연구의 배경과 목적에 대한 내용을 기술하며 2장은 기존에 연구된 비문 인식 기술의 선행연구에 대해 살펴보고 객체 인식 모델의 기술적 동향 및 기존 연구와의 차별화 요소를 도출하였다.

3장에서는 실험환경과 ORB에 BEBLID Descriptor에 대하여 설명하며 ORB에 BEBLID Descriptor의 실험 결과에 대하여 기존의 인식기법과 제안 기법의 인식률을 비교 기술한다. 4장 결과 분석에서는 실험 결과를 정량적으로 분석한 내용을 기술하며 5장 결론에서는 연구의 분석 결과를 기반으로 본 논문의 종합적인 정리와 한계점을 기술하였다.



[그림 1-2] 연구 과정

Ⅱ. 이론적 배경

2.1 객체 인지 기반 기술

2.1.1 객체 인지 기술 개요

영상 내 물체를 식별하는 과정으로 객체 검출(Object Detection)과 객체 인식(Object Recognition)으로 나눌 수 있으며 객체 검출은 이미지에서 특정 클래스의 물체(예: 사람, 자동차, 동물 등)의 위치를 감지하는 가장 근본적인 기술이며 객체 추적, 분할, 캡션 등의 기반 기술로 응용된다(Zou et al; 2023). 객체 인식은 인간이 사진이나 영상을 바라볼 때, 쉽게 사람, 객체, 장면, 시각적 요소들을 찾는 기술로 목표는 컴퓨터가 자연스럽게 인지하고 행동하는 것을 가르치는 것으로, 이미지 내에 무엇을 가지고 있는 지에 대한 이해 수준을 얻는 것이다(Lowe, D., 2004).

객체 인식에 대한 연구는 오랫동안 진행되었으며 이는 모델의 딥러닝 과정 또는 가중치 업데이트 기법을 통해 진행되었다. 주요 과정은 입력층-은닉층-출력층을 통해 예측값을 내는 방법으로 순전파(Forward propagation)에서 데이터를 입력받은 모델은 무작위로 초기화된 가중치를 이용해 예측값을 출력하게 된다. 예측값과 정답 사이의 차이를 정의하는 손실 함수(Loss Function)를 이용해 입력에 대한 손실을 계산하며 출력층에서 입력층으로 거슬러 올라가는 역전파(Backward propagation)과정에서 최적화 알고리즘은 기울기(Gradient) 값을 이용해 앞서 구한 손실을 최소화하는 방향으로 모델의 가중치 값을 단계적으로 수정한다. 이러한 과정을 반복해서 거치는 딥러닝 모델은 여러 번 반복해서 데이터를 점진적으로 학습하게 되어 최적 객체를 인지하게 된다.

2.1.2 객체 인지 주요 기술

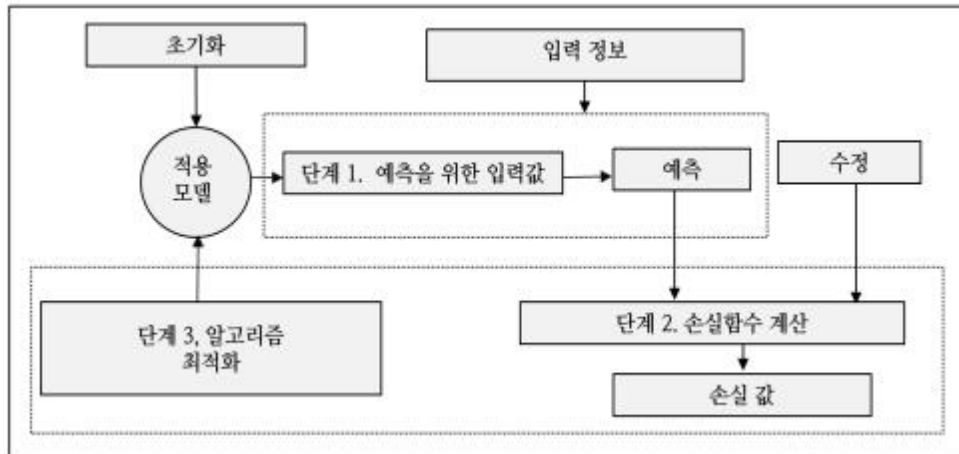
객체 인식은 데이터의 객체 위치가 제각기 다르거나 그 촬영 각도가 다르면 객체의 인식 정확도가 낮아질 수 있으며 사진에서 객체 영역을 찾아 동일한 형태의 객체를 추출하는 과정이 선행되어야 한다. 이는 시스템에 입력된 이미지에서 객체 영역을 찾아(Object detection) 객체의 특징을 나타내는 점(Object alignment)의 특징점을 이용해 객체 영역을 동일한 형태와 크기로 변경(Normalization)하는 작업을 수행한다.

객체 인식 모델은 수집된 객체 데이터로부터 정제화 과정을 거쳐 획득한 이미지의 객체 표현 특징 벡터(Feature vector)를 학습하며 입력된 이미지의 특징 벡터 간 유사도를 비교하는 방식으로 식별을 수행한다. 대표적인 객체 인지 기술로 딥러닝 기술(Deep structured learning)이 있으며 이는 심층 학습 방식으로 다량의 데이터나 복잡한 자료들 속에서 핵심적인 내용 또는 기능을 요약하는 추상화 작업을 통하여 학습을 통하여 객체를 인지하는 기계 학습 알고리즘의 집합으로, 큰 틀에서 사람의 사고방식을 컴퓨터에게 가르치는 기계학습의 한 분야라고 이야기할 수 있다(Bengio, Y. et al, 2013).

딥러닝 기술들은 객체 인식을 함에 있어 컨볼루션 뉴럴 네트워크와 같은 딥러닝 모델(CNN)들을 사용하며 객체를 식별하기 위해서 자동적으로 객체의 내재 특징을 배우게 되며 객체 인식을 딥러닝을 이용하여 수행한다.

대부분의 딥러닝 애플리케이션들은 전이 학습 방식을 이용하며 이는 기존에 학습된 모델을 적절하게 튜닝하는 과정을 이용하여 이전에 알려지지 않은 분류들을 포함하는 새 학습 데이터를 생성한다. 이 방식은 기존에 수많은 이미지들로 모델을 통해 이미 학습되었기 때문에 시간이 절약되며 빠른 결과를 도출해 낼 수 있다. 딥러닝 중 가장 중요한 것은 강력한 GPU와 많은 양의 라벨화된 데이터이며 많은 데이터에서 좋은 인식 및 매칭 결과를 도출하며 모델을 학습하는 시간을 단축한다.

딥러닝 가중치 업데이트 모델의 훈련 과정은 [그림 2-1]과 같이 최적화 과정을 수행한다.



[그림 2-1] 객체 인식 딥러닝 절차

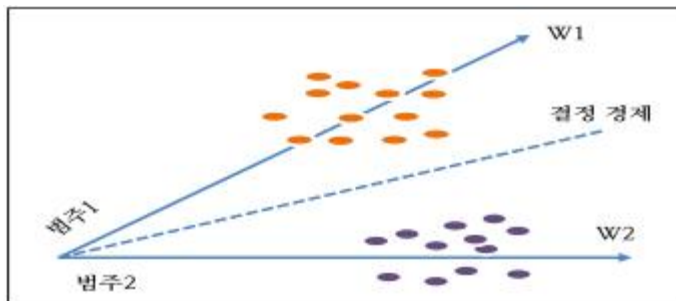
객체 인식 딥러닝 과정은 데이터를 입력받아 무작위로 초기화된 가중치를 이용해 예측값을 출력 한후 예측값 차이를 정의하는 손실 함수를 이용해 입력에 대한 손실을 계산하는 순전파(Forward propagation) 과정을 수행한다. 이후 기울기 값을 이용해 앞서 구한 손실을 최소화하는 방향으로 모델의 가중치 값을 수정하는 역전파(Backward propagation) 과정을 진행한다. 딥러닝 모델은 이 과정을 여러번 반복해서 데이터를 점진적으로 학습하게 된다.

객체 인식과 객체 탐지는 물체 식별을 하는데 유사한 기술이다. 그러나 이 둘은 실행 함에 있어 객체 탐지는 이미지 내의 물체 존재를 찾는 과정이며 객체 인지는 탐지된 객체의 종류를 인식하는 과정으로 차이가 있다. 딥러닝의 경우, 이미지를 식별하는 것뿐만 아니라 이미지의 위치도 찾는 객체 탐지도 수행하며 이는 복수의 물체가 식별되는 이미지에서 객체의 종류와 위치를 동시에 인지하게 된다.

2.1.3 객체 인지 학습 손실 함수

객체 인식을 학습하기 위한 손실 함수는 소프트맥스 손실 함수(Softmax loss function), 거리 기반 손실 함수(Distance-based loss function), 앵글러 마진 기반 손실 함수(Angular margin based loss function) 등 크게 세 종류로 나뉘볼 수 있으며 이를 통하여 객체를 인식한다. 소프트맥스 함수는 범주 수만큼의 차원을 갖는 입력 벡터를 받아서 확률로 변환하는 방식으로 다 범주 분류기에서는 출력값을 제대로 해석하고 다른 출력 노드와의 상대적인 크기를 비교하여 각 출력 노드를 0~1 사이로 제한해 이를 합한 값을 1로 만든다.

관련 함수로 교차 엔트로피(Cross entropy)가 있으며 소프트맥스 확률의 분포와 정답 분포와의 차이를 나타낸다. 이를 기본으로 해서 딥러닝 모델 학습 과정을 거치며 이는 손실을 최소화하는 방향으로 모델의 각 파라미터를 업데이트하는 과정을 반복하게 된다. 교차 엔트로피가 소프트맥스 확률 분포와 정답 분포 사이 오차를 계산하며 소프트맥스 확률값을 이용한다는 점에서 소프트맥스 손실 함수라고 한다.



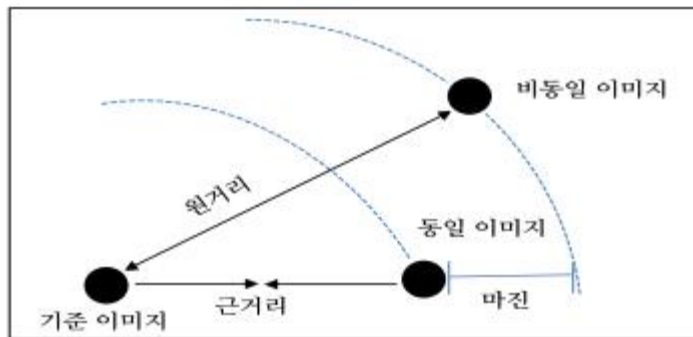
자료 출처 : <https://tech.kakaoenterprise.com/63>

[그림 2-2] 소프트맥스 손실함수

소프트맥스 손실 함수값을 지속적으로 최소화하는 학습을 통해 인식모델은 설정된 공간에 있는 일치하는 객체 간의 두 특징 벡터를 더 근접하

게, 일치하지 않는 비동일 객체의 두 특징 벡터는 더 멀게 표현한다.

소프트맥스 손실 함수는 국부적인 최적 지점으로 간단하게 수렴하므로 많은 데이터의 특징 공간을 효율적으로 학습하기가 어려우며 전체적인 최적화를 고려하지 못하고 있다. 거리 기반의 객체 인지 기술은 소프트맥스 손실 함수와 유사하게 객체를 인식하는 특징 공간에 동일 객체의 두 특징 벡터를 더 가깝게, 비동일 객체의 두 특징 벡터는 더 멀게 표현하는 학습 방식으로 동일한 개념이다. 차이점은 특징 벡터 간의 거리를 학습에 활용하는 것으로 대비손실 함수와 트리플렛(triplet) 손실 함수를 통해 인지한다. 대비 손실 함수는 두 객체 이미지의 쌍을 구성해 두 특징 벡터 간의 거리를 계산하며 여기서 손실 값은 객체의 두 벡터 간 거리가 멀면 커지고, 비동일 객체의 두 벡터 간 거리가 가까우면 커지게 된다. 이에 학습 모델은 이 손실 값을 최소화하는 반복 학습을 통해 동일 객체에 해당하는 두 벡터를 가깝게, 비동일 객체의 두 벡터를 더 멀게 표현하게 된다.



자료 출처 : <https://tech.kakaoenterprise.com/63>

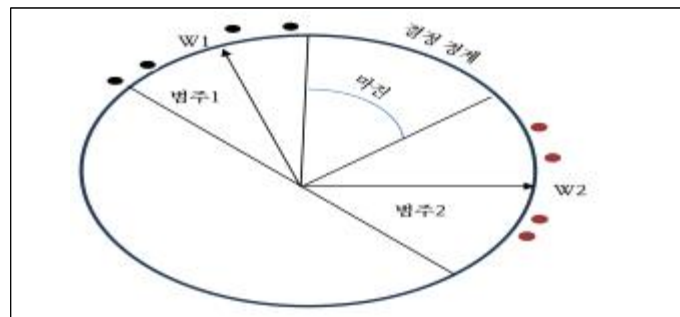
[그림 2-3] 거리 기반 손실함수

거리 기반 손실함수 계산 방식은 동일 객체에 해당하는 두 벡터 거리와 비동일 객체의 두 벡터 거리가 개별적으로 학습하기 때문에 동일 객체 간의 벡터 거리가 비동일 객체 사이 벡터 거리보다 상대적으로 가까워져야 함을 고려하지 못하는 문제점을 내포하고 있다.

따라서 이러한 문제점을 해결하기 위한 트리플렛 손실 함수는 범주가 같은 벡터 간의 거리와 범주가 다른 벡터 간의 거리의 상대적 관계를 고려하여 이를 해결한다. 기준이 되는 이미지 a, 동일 객체 이미지 b, 그리고 비동일 객체 이미지 c로 구성된 트리플렛(a, b, c)을 구성하며 트리플렛 손실 함수는 동일 객체인 a와 b의 벡터 간 거리를 줄이는 동시에, 비동일 객체인 a와 c의 벡터 간 거리를 넓게 산정한다. 즉, 두 거리의 차($|a-c| - |a-b|$)가 임의로 정한 마진값보다 크도록 하며, 트리플렛 손실 함수로 학습된 인식 모델은 대비 손실 함수와 비교했을 때 더 높은 추론 성능을 보인다.

그러나 트리플렛을 구성하는 방식에 따라 범주 수가 많을수록 효과적으로 구성하기 어려우며 비문과 같이 분류해야 할 범주가 많은 객체에서는 효과적인 객체 인식이 어려운 한계점을 지니고 있다(장도화, 권경석, 김중곤, 양가영, 김종복, 2020). 근래의 객체 인식 기술은 앵글러 마진을 추가한 소프트맥스 기반 손실 함수를 이용해 서로 다른 객체 간 거리를 충분히 넓히는 방향으로 연구되고 있다.

이처럼 손실 함수는 객체 인식 모델에게 서로 다른 객체의 특징점을 구분하여 최적의 특징 공간을 생성하는 방법으로 이를 활용한 객체 인식 연구는 수천 개의 범주를 분류하는 데 최적화된 범용 모델을 사용해서 데이터 정보를 효과적으로 처리하여 인지하도록 관련된 기술과 방법을 제공한다(Mei Wang & Weihong Deng, 2018).



자료 출처 : <https://tech.kakaoenterprise.com/63>

[그림 2-4] 앵글러 마진 기반 손실 함수

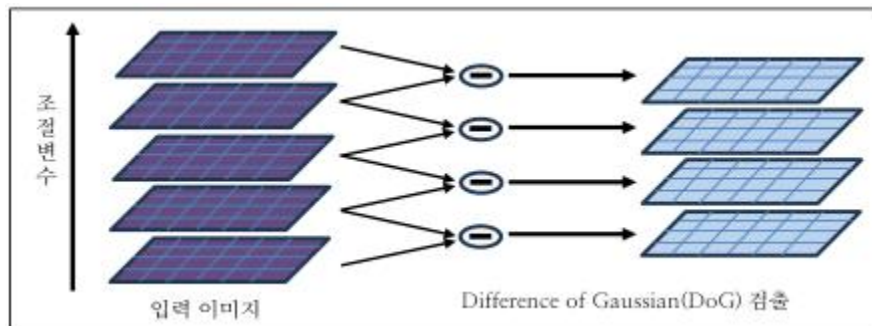
2.2 객체 인식 알고리즘

2.2.1 SIFT(Scale Invariant Feature Transform)

객체 이미지의 스케일 변화에 민감한 문제를 해결한 알고리즘인 SIFT는 다양한 이미지 스케일 공간을 구성하고 Difference of Gaussian(DoG) 검출기를 사용하여 객체에서 키포인트의 차이를 분석하여 특징점 모서리를 찾는 방식으로 [그림 2-5]에서 간략히 도식하였다.

Lowe, D. 의 SIFT 알고리즘 연구에서 객체 이미지 I 는 x 와 y 좌표로 구성하며 이를 DoG의 컨볼루션(Convolution)을 이용하여 특징점을 검출하였다. 각 키포인트에 대해 디스크립터의 표시는 특징점의 기울기 크기와 근처 픽셀부터 상대적인 방향으로 표현되며(Lowe, D., 2004)). 이미지의 상대적인 방향 계산을 위하여 도출 과정이 복잡하고 도출시간이 필요하다. 관련된 계산식은 아래와 같고 "G"는 가우스 함수를 나타낸다.

$$(D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y)$$



자료 출처 : https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html

[그림 2-5] SIFT 알고리즘

SIFT는 가장 널리 사용되고 널리 사용되는 특징 추출 기술 중 하나로 크기, 회전 및 조명 변경에 변하지 않는 이미지의 로컬 기능을 감지하고

설명하는 데 사용된다. SIFT는 이미지에서 핵심 포인트를 감지하고 일련의 고유한 기능을 사용하여 설명하는 방식으로 SIFT 알고리즘에는 스케일 공간 극값 검출, 키포인트 현지화, 방향 할당 및 특징 기술자 추출을 포함한 여러 단계를 포함하여 사용한다.

SIFT 알고리즘은 이미지에서 특징점을 추출하는 대표적인 알고리즘 중의 하나로 가우시안 필터를 이용하여 이미지의 크기가 변하거나 회전하여도 같은 위치에서 특징을 추출하는 특성을 가지고 있다.

SIFT 특징은 객체 검출, 객체 추적, 이미지 정합, 패턴 인식 및 3D 모델링과 같은 다양한 응용 분야에서 활용된다. SIFT 특징 추출의 주요 특징으로 첫 번째, 크기 불변성(Scale-Invariant)이 있으며 이는 이미지 내에서 동일한 물체나 특징이 다양한 크기로 나타날 때도 식별할 수 있다. 두 번째, 회전 불변성(Rotation-Invariant) 있으며 물체나 특징이 회전되어도 식별 가능하다. 세 번째, SIFT는 이미지의 로컬 영역에서 특징을 추출하며, 이로써 이미지 내의 여러 객체나 패턴을 독립적으로 다룰 수 있는 로컬 특징 추출이 가능하다. 네 번째, 각 특징 포인트는 방향, 크기, 그래디언트 정보, 및 주변 픽셀들의 히스토그램을 포함한 높은 차원의 특징 벡터로 표현을 할 수 있다. 반면에 SIFT는 많은 연산량으로 인해 속도가 느려서 실시간 처리 또는 많은 양의 영상에서 특징점을 추출하기에는 어려움이 있다.

이러한 SIFT의 문제점을 해결하기 위해 SURF, FAST, 그리고 AGAST 등이 개발되었으며 SIFT 알고리즘의 가우시안 필터와 박스 필터를 이용하여 특징점을 추출한다(Tareen, S. & Saleem, Z., 2018). 이는 이미지 피라미터 대신 필터의 크기를 변화시키는 방식으로 SIFT 알고리즘에 비해 정확도는 떨어지나 속도 개선을 위해 누적 영상과 박스 필터를 사용함으로써 연산량이 줄어들도록 한 알고리즘이다.

SIFT는 이미지에서 특징 포인트를 검출하고, 특징 디스크립터로 설명하는 과정으로 이루어지며 이미지 처리 및 컴퓨터 비전 분야에서 고전적이고 많이 사용되는 기술 중 하나이다.

2.2.2 SURF(Speed-Up Robust Features)

SURF는 Herbert Bay, Tinne Tuytelars, Luc Van Gool에 의해 처음 연구되었으며 계산 횟수를 줄이기 위해 기술자를 생성하는 과정에서 적분 이미지를 사용하여 핵심 포인트를 감지하는 알고리즘이다. SURF는 특정 키포인트에서 방향에 대한 전방위 각도가 추출되고 기울기 정보가 하위 영역에서 벡터값으로 나타낸다는 점에서 SIFT와 비슷하며 객체 인식, 영상 등록, 분류 또는 3D 재구성 등의 작업에 사용할 수 있다(Bay, H, Tinne, T. & Luc, V. 2008).

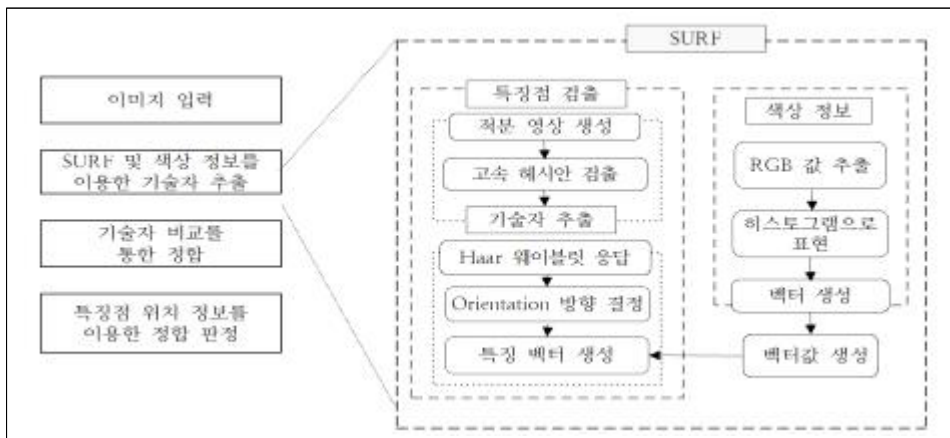
SURF의 특징점 탐지는 피라미드 구조의 스케일 공간 생성(scale-space of pyramid), 후보점 추출, 비 최대치 억제 기법(Non-maximum suppression)의 세 단계를 거쳐 수행한다. SURF는 미리 계산된 적분 이미지를 사용하여 3개의 정수 연산으로 계산할 수 있는 Hessian blob 검출기 행렬식의 정수 근사치를 사용하며 기술자는 관심 지점 주변의 Haar wavelet 응답의 합을 기반으로 산정한다. 이를 통해 SURF 기술자는 물체, 사람 또는 얼굴을 찾고 인식하며, 3D 장면을 재구성하고, 물체를 추적하고 관심 지점을 추출하는 데 사용되며 다중 스케일 영상에서 특징을 찾음으로써 스케일 불변성을 달성한다(Bay, H, Tinne, T. & Luc, V. 2008).

또한 다중 해상도 피라미드 기법을 사용하여 Pyramidal Gaussian 또는 Laplacian Pyramid 모양으로 원본 이미지를 복사하여 크기는 같지만 대역폭은 줄어든 이미지로 변환한다. 이렇게 하면 Scale-Space라고 하는 원본 이미지에 특별한 blurring 효과를 얻을 수 있으며 관심 지점의 스케일 불변성을 유지한다.

SURF 알고리즘은 SIFT보다 몇 배 더 빠르며, 다른 이미지 변환에 더 강력한 특징을 가지고 있어 주로 객체 인식, 영상 등록, 분류 또는 3D 재구성 등에 사용한다. 특히 SURF는 3D 장면을 재구성하고, 물체를 추적하고 관심 지점을 추출하는 데 사용이 가능하여 물체, 사람 또는 얼굴을 찾

고 인식하며, 이미지가 회전하여도 특징점이 불변하는 특징을 가지고 있다.

SURF의 추출 이미지는 다중 해상도 피라미드 기법을 사용하여 좌표로 변환되어 Pyramidal Gaussian 또는 Laplacian Pyramid 모양으로 변환되어 크기는 같지만 대역폭은 줄어든 효과를 얻는다. 이렇게 하면 Scale-Space라고 하는 원본 이미지에 특별한 blurring 효과를 얻을 수 있으며 관심 지점이 scale invariant을 유지할 수 있게 된다(Alahi, A. Ortiz, R. & Vanderghenst, P., 2012).



[그림 2-6] SURF 알고리즘 구조

[그림 2-6]은 SURF 알고리즘의 핵심인 기술자 추출 부분을 설명하고 있으며 SIFT와 비슷한 성능을 나타내고 있으나 고속 헤시안 검출 기법을 사용하여 특징점 추출 수행 시간이 SIFT에 비해 훨씬 빠르다는 장점이 있다. 또한 특이점을 찾고 크기와 방향을 비교하여 매칭을 시도하다 보니 이미지의 크기가 달라져도 인식에는 문제가 되지 않는다. 그러나 SURF는 회전 불변한 특징 보장을 위해서 추출한 특징점 간의 위치 정보를 고려하지 않는 단점을 가지고 있다. 즉, SURF는 물체의 기울기, 크기, 방향, 밝기 등의 차이에 영향이 적으며 이미지의 크기가 다르더라도 매칭이 가능하나 회전성에는 상대적으로 약한 객체 인식 알고리즘이다.

2.2.3 BRIEF(Binary Robust Independent Elementary Features)

BRIEF는 Binary Descriptor의 한 종류로 객체의 회전 변화에 강한 인식 알고리즘(Rotational BRIEF)을 사용하며 이는 회전하여도 데이터의 불변성을 보장하기 위해서 정해진 Keypoint의 방향에 따라 기술자를 조정한다. 기술자의 조정은 특징점 주변 영역의 픽셀을 다른 픽셀과 명암을 비교해 밝은 부분을 찾아 이진 형식으로 저장하는 방식으로 진행된다.

BRIEF 알고리즘은 이미지를 컨벌루션 처리를 위해 가우시안 커널을 사용하며, 이는 피쳐 중심 주변의 가우스 분포를 통해 첫 번째와 두 번째 구간을 비교해 관련 픽셀을 한 쌍으로 인지하며 두 개의 픽셀을 하나의 그룹으로 묶는 방식을 사용한다.

특정된 위치(x_i, y_i)에서 n 개의 feature set에 대해 BRIEF 알고리즘으로 이진 테스트를 진행한다면 아래와 같은 행렬이 필요하다.

$$S = \begin{pmatrix} x_1, \dots, x_n \\ y_1, \dots, y_n \end{pmatrix}$$

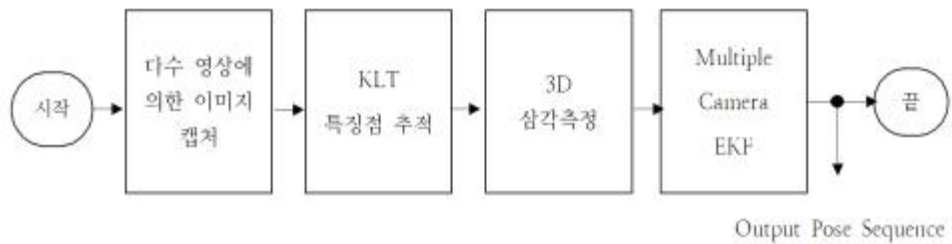
이때, 이전에 구한 패치(Patch)의 방향 정보(Orientation) ϕ 로 회전형렬 R_ϕ 를 생성하여 S 를 회전시켜 다음과 같은 값을 구한다.

$$S\phi = R_\phi S$$

$S\phi$ 를 사용하여 방향 정보 ϕ 에 따라 steered BRIEF를 정의할 수 있으며 회전된 비문 정보에 대하여 초기 데이터의 회전값 불변 원리에 의해 비교값을 매칭시켜 일치점을 찾아내는 방식이다.

BRIEF 알고리즘은 객체의 영상에서 중요 특징점의 0과 1의 이진 요소(Binary Description)를 특징점 주변으로 가우시안 분포를 가지는 비교 쌍과 비교를 통해 기술자를 추출한다. 선택된 영역(ROI)의 Keypoint(k)에 대하여 x 와 y 의 픽셀값의 Binary를 서로 비교하여 128bit의 이진 기술자를 생성하게 된다. BRIEF는 descriptor를 찾지 않아도 이진 문자열을 직접 찾

는 방식으로 서로 다른 이미지의 특징점을 추출하여 상호 연관성이 있는지 분석한다. BRIEF는 Descriptor 계산과 매칭에 빠른 방법으로 큰 회전이 없는 한 SIFT, SURF 알고리즘에 비해 높은 인식률을 나타낸다.



[그림 2-7] BRIEF 알고리즘

[그림 2-7]은 BRIEF 알고리즘 처리 절차로 빠른 처리 속도를 목표로 랜덤하게 선택된 픽셀 쌍의 차이를 계산하여 binary string으로 변환하는 Descriptor Extraction 방식을 사용한다. 일르 위해 특징점 추적 및 삼각측정 방식을 사용하며 수학적 측면으로 주로 Hamming 거리를 사용하여 아래와 같이 수행한다.

$$Hamming Distance = \sum_{i=1}^n |a_i - b_i|$$

BRIEF 알고리즘은 생성과 비교 연산을 최소화하여 인식 속도를 향상했으며 Binary Bit를 이용하기 때문에 메모리 측면에서 효율적이다. 그러나 이미지가 회전하거나 크기가 크게 변화했을 경우 인식이 힘든 문제점을 내포하고 있다(Calonder, M., 2010). 이러한 문제점을 개선하기 위하여 rBRIEF(Rotated BRIEF) 알고리즘이 등장하였으며 이는 BRIEF에서 이미지의 회전이 크게 변화하더라도 특징점의 인식에는 큰 차이를 나타내지 않게 하는 개선된 방식이다.

2.2.4 ORB(Oriented FAST and Rotated BRIEF)

“ORB : an efficient alternative to SIFT or SURF”는 2011년에 발표된 이론으로 실시간 이미지 객체 인식 기술의 새로운 기술로 등장한 알고리즘이다. 객체인지 기술은 정확한 이미지의 분석이 중요하며 ORB 알고리즘은 Feature Accelerated Segment Test(FAST)의 기능과 BRIEF(Binary Robust Independent Elementary Features) 알고리즘, 해리스 코너 검출 알고리즘을 결합한 방식을 사용하였다(Rublee, E. et al 2011).

ORB 알고리즘은 FAST 알고리즘을 사용해 특징점을 검출하며 FAST 알고리즘은 Feature detection을 통해 Key Point를 추출하고 Descriptor를 계산하는 방식이다. 즉 ORB 알고리즘은 Key point와 Descriptor를 산출하여 ORB feature를 최종 결과물로 내는 알고리즘으로 로스텐(Rosten)과 드리먼드(Drummond)가 제안한 피쳐 검출기 알고리즘이다. 이는 픽셀 P와 픽셀 주변의 점을 비교하는 방식이며 픽셀 P가 주변 픽셀에 임계값을 적용해 어두운 픽셀, 밝은 픽셀, 유사한 픽셀로 분류하여 분석하며 우수한 특징점은 주변 픽셀이 연속적으로 어둡거나 밝아야 하며 이 연속성이 절반 이상이 되어야 하는 개념이다.

FAST 알고리즘은 해리스 코너 검출 알고리즘을 적용해 코너뿐만 아니라 가장자리에도 반응하는 문제점을 해결하였으며 최상위 특징점만 추출하는 방식이다. 또한 스케일 공간 검색을 수행하기 위해 이미지 피라미드를 구성하며 스케일 크기의 변화에 따라 피쳐 주변 박스 안의 강도의 분포에 대해 X축과 Y축을 기준으로 1차 모멘트를 산출한다.

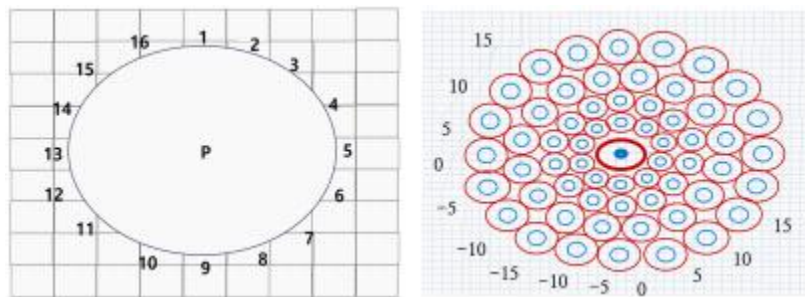
1차 모멘트는 그레이디언트의 방향에 따라 피쳐의 방향을 지정할 수 있으며 해당 방향에 대해 피쳐 벡터를 계산하며 이때 피쳐는 방향 정보를 포함한 회전 불변성을 갖고 있다.

ORB 알고리즘은 기존 객체 인지 방식의 대표적 문제점으로 영상의 크기가 달라지는 경우 특징점으로 검출되지 않을 수 있는 단점을 개선하여

FAST 코너 검출 방법을 통해 특징점을 추출한 뒤, 특징점의 방향성분을 계산하고 BRIEF로 이진 기술자를 사용하여 특징점의 방향 정보를 정확한 값으로 계산하여 객체 인지 성능을 향상시켰다. 즉 ORB 알고리즘은 객체의 크기에 따라 특징점 추출이 어렵고 여러 스케일의 영상을 이용하며 영상이 회전하였을 경우 인지율이 저하되는 문제점을 해결한 방식으로 Rotation invariant와 Resistant to noise를 결합한 방식으로 단순히 Fast와 BRIEF를 두 알고리즘을 개선한 새로운 알고리즘이다.

초기 ORB에서는 다중 스케일에 대한 특징점 추출을 위해서 멀티스케일 이미지 피라미드(Multiscale image pyramid)를 적용하였다. 이는 객체의 특징점이 내포한 주변 요소에 따라 크기 불변성을 가질 수 있도록 하기 위하여 각각의 특징점을 피라미드의 형태로 인지하였으며 구성요소는 이전 크기에 대해 축소, 확대 등 여러 크기에 대하여 특징점을 미리 추출함으로써 크기에 관계 없이 특징점을 인지토록 한 방식이다.

FAST는 밝기값을 특정 반경 내 주변 픽셀과 비교하여 단순한 밝기 비교를 통해 매우 빠른 속도로 특징점을 추출하며 임의의 픽셀 'p' 주위로 16개의 주변 픽셀 영역을 설정하며 'p'와 각 픽셀들의 밝기 차이를 비교하여 'N'개 이상의 연속된 픽셀이 'p'보다 밝거나 어두우면 'p'는 특징점으로 판명되는 알고리즘으로 [그림 2-8]에 나타내고 있다.



자료 출처 : https://docs.opencv.org/3.4/df/d0c/tutorial_py_fast.html

[그림 2-8] FAST 특징 추출 알고리즘

특징 추출 알고리즘은 비교 쌍을 가우시안 분포로 랜덤하게 선택하고 비교 쌍들 간의 밝기 비교를 통해 기술자를 구성하며 이진 테스트(Binary test)를 통해 아래 식과 같은 n 개의 비트로 구성된 n -차원의 이진 문자열(binary string) des_n , BRIEF가 추출된다.

$$r(p; x, y) := \begin{cases} 1 : I(x) < I(y) \\ 0 : I(x) \geq I(y) \end{cases} \quad (1)$$

$$des_{n, \text{BRIEF}}(p; i) = 2^{i-1} r(p; x_i, y_i), \quad (2)$$

where $1 \leq i \leq n$

* 'p': 31x31 하위 윈도우인 이미지 패치(image patch)

$I(x), I(y)$: 이미지 패치 p 에서 각각 x 와 y 위치 픽셀의 밝기

FAST는 다른 코너 검출 알고리즘에 비해 빠르게 동작하나 데이터 이미지에 노이즈가 많을 경우 제대로 된 결과가 나오지 않는 단점이 있다.

ORB 알고리즘은 객체의 특징점을 인지하기 위해 이미지 모멘트(Image moment)를 적용하며 이진 기술자에 키포인트의 주 방향 각도를 추가하여 이미지 회전에 강한 특징이 있어 객체의 방향성을 할당할 때 분포 상태에 의해 특징점을 인지하도록 한다. ORB 알고리즘의 특징점은 클래스 ID(class_id)를 포함한 좌표(pt)와 지름(size), 각도(angle), 옥타브(octave), 응답(response)을 포함한다. 클래스 ID는 클러스터링한 객체 ID로 특징점을 저장공간에 저장할 때 객체를 구분하기 위함이며 좌표는 특징점의 위치 정보, 지름은 특징점의 주변 영역을 의미한다. 각도는 특징점이 위치하는 방향 정보이며 옥타브는 특징점을 추출한 피라미드의 스케일을 의미하며 응답은 피처가 존재할 확률을 나타낸다. 이를 포함한 기술자는 각 특징점을 설명하기 위한 2차원 배열로 표현되며 이 배열은 두 특징점의 동일 여부를 판단할 때 사용된다(탁기현, 방효충, 2015). 즉, ORB 알고리즘은 SIFT 알고리즘과 SURF 알고리즘을 대체하기 위해 OpenCV Labs에서 개발됐으며 속도 또한 빨라진 방식이다.

2.2.5 BEBLID(Boosted Efficient Binary Local Image Descriptor)

2016년 옥스포드 대학에서 "BEBLID : Boosted Efficient Binary Local Image Descriptor"라는 제목의 연구 논문에서 소개되었으며 이는 이미지 특징 추출 및 일치에 사용되는 컴퓨터 비전 알고리즘으로 기존의 SIFT 및 SURF와 같이 계산 비용이 많이 들고 많은 양의 메모리를 필요로 하는 기존 로컬 기능 기술자의 한계를 해결하였다(Lago, S., 2020). 기존 ORB 알고리즘보다 매체 인식률을 향상하여 이미지 일치 정확도를 높이고 실행 시간을 줄이는 새로운 기술자(Descriptor)인 BEBLID를 본 연구에 적용하였으며 ORB 키포인트의 스케일의 인수값을 조정하여 ORB의 기술자를 개선하는 효과를 검증하였다(Shuaishuai, P., 2022).

BEBLID 알고리즘은 컴퓨터 비전 응용 프로그램에서 효율적인 지역 이미지 특징의 매칭을 수행하는 데 사용되며 이전에 사용되던 실수값 기반 알고리즘 매칭을 개선하여 매칭의 효율성과 정확성을 높였다. 주요 특징으로 BEBLID는 효율적으로 학습된 이진 이미지 기술자를 사용하며 AdaBoost를 활용한 개선된 학습자 훈련 방법을 사용하여 더 나은 인식률을 지향하며 데이터 불균형을 해결하기 위한 이진화 방식을 사용하여 모든 약한 학습자가 강한 학습자 조합에서 동일한 가중치를 갖도록 하고, 매칭 및 검색 작업에서 발생하는 비대칭성을 해결하기 위해 데이터 불균형을 고려하여 디스크립터를 이진화한다(Wang, N., 2020).

즉 이미지의 작은 부분을 일련의 0과 1로 설명하며 같은 이미지 부분을 다른 시점에서 감지할 때 상당히 유사하게 유지되며 이러한 이진 문자열은 이미지의 지문과 같은 효과를 나타낸다. ORB, LATCH 또는 BinBoost와 같은 유사한 기술자와 비교했을 때, BEBLID는 더 빠르고 정확하게 특징점을 설명하며 하나의 이미지를 설명하는 데 걸리는 시간은 2 밀리초 미만으로 빠른 성능을 나타낸다.

본 연구에서는 ORB 키포인트를 감지하고 BEBLID를 사용하여 설명한

다음 OpenCV BFMatcher를 사용하여 두 이미지 간의 일치하는 특징을 출력하는 방식을 사용하였으며 아래와 같은 프로그램을 사용하였다.

```
import cv2

# Load an image
image1 = cv2.imread("image1.jpg")
image2 = cv2.imread("image2.jpg")

# Detect ORB keypoints
orb = cv2.ORB_create()
keypoints1, descriptors1 = orb.detectAndCompute(image1, None)
keypoints2, descriptors2 = orb.detectAndCompute(image2, None)

# Create BEBLID descriptor
beblid = cv2.BEBLID_create()
beblid_descriptors1 = beblid.compute(image1, keypoints1)
beblid_descriptors2 = beblid.compute(image2, keypoints2)

# Match features using BFMatcher
bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
matches = bf.match(beblid_descriptors1, beblid_descriptors2)

# Print matched features
for match in matches:
    print(f"Image1    Keypoint:    {match.queryIdx},    Image2    Keypoint:    {match.trainIdx}")
```

BEBLID의 핵심 아이디어는 부스트 분류자(Boosted classifiers)를 사용하여 기능의 하위 집합을 선택하여 이용하는 것으로 BEBLID는 회색조 이미지에서 작동하며 이미지가 작은 패치로 분할되고 각 패치에서 피처가 독립적으로 추출되는 패치 기반 접근 방식을 사용하며 실수값 벡터를 사용하는 기존 기능 기술자보다 메모리 효율적인 이진 기능 기술자를 추출하여 효율적으로 이용하며 가장 빠른 알고리즘인 ORB보다 SIFT에 가까운 정확

도와 더 나은 계산 효율성을 제공한다.

SIFT는 가장 잘 알려진 특징 검출 및 설명 알고리즘으로 많은 컴퓨터 비전 작업에서 좋은 성능을 발휘하기 때문에 널리 사용된다. 그러나 특정 상황에서 실시간 성능을 달성하기 위해 GPU를 사용해야 하는 것은 계산적으로 상당히 까다롭다.

SIFT 속도를 높이기 위해 SURF, BRIEF, BRISK, ORB, FREAK, BOLD 와 같은 다양한 알고리즘의 기술자가 등장했으며 이진 접근법은 효율적인 이진 로컬 이미지 설명자인 이진값 기술자를 생성한다.

본 연구에서는 정확성 측면에서 최첨단 기술과 계산 요구 사항 측면에서 가장 효율적인 이진 이미지 기술자인 BEBLID를 이용하며 기존 연구에서 가장 빠른 설명자인 ORB보다 정확성 및 매칭 속도가 빠르다는 것을 증명했다. 이는 적분 이미지로 계산된 그레이값 차이를 기반으로 하는 매우 효율적인 이미지 기능을 사용하기 때문이다.

이진 이미지 기술자인 BEBLID의 스케일 계수는 이미지 피라미드를 설정하며 스케일 계수를 너무 크게 지정하면 특징점의 매칭 확률을 떨어뜨리고 반대로 스케일 계수를 작게 지정하면 더 많은 피라미드 레벨을 구성해야 하므로 연산 속도가 느려지게 되어 적절한 스케일 계수 선정이 중요한 부분이다. 이 기술자는 다양한 이미지 검출기에서 특징점을 설명할 수 있으며, `scale_factor` 매개변수를 조정함으로써 다양한 크기의 특징점을 처리할 수 있다. BEBLID는 같은 다른 이진 기술자보다 효율성을 유지하면서 성능을 크게 향상시킨 방식으로 실시간 이미지 특징 매칭에 적합하며 AdaBoost를 사용하여 개선된 약한 학습자(Weak-learner) 훈련 방법을 적용하여 더 나은 지역적인 특징을 생성하는 방식이다. 본 연구에서는 ORB 검출기를 사용하여 특징점을 감지하고, BEBLID의 Descriptor를 이용하여 특징점을 설명하는 특징을 이용하여 2개의 알고리즘을 조합한 최적의 Descriptor를 도출한다.

2.3 비문 기반 객체 인지 기술

2.3.1 비문 정보의 정의

오늘날 생체인증 기술(Biometrics)이란 인간의 얼굴 등 신체 일부의 생리학적 특징 또는 움직임 등 행동상의 특성을 기반으로 독특한 특징을 추출하여 분석하며 이를 수치화하여 데이터화 후 이를 기반으로 개개의 확인을 위한 기술로, 인간의 자연적 또는 인위적 특징을 인공지능 등 자동화된 프로그램을 통해 개인의 확인에 이용하는 기술 분야로 다양하게 사용된다.

생체인증 기술은 개인의 식별 또는 확인을 위해 개인마다 다르게 가진 얼굴, 지문, 홍채, 음성, 지문 등 생체정보를 추출하여 분석하고 이를 저장 장치에 등록시키고, 이후 생체 입력 장치를 통해 판별이 필요한 개인의 생체정보 특징을 등록된 정보와 매칭시켜 비교하여 확인함으로써 개인 식별의 수단으로 활용하는 것이다. 생체인식정보는 특정한 개인을 인증하거나 식별하기 위해 사용되며 인증 혹은 식별을 위한 목적이 아닌 연령, 성별, 감정과 같은 개인에 관한 특징을 알아보기 위한 일반적 생체정보들로 구성된 정보로 정의하고 있다. 생체 인식 기술은 유일한 신체의 특징을 이용하며 주로 지문, 얼굴, 정맥, 홍채, 음성, 서명 등을 활용하며 손상되거나 민감하지 않으며 정확도가 높은 요소를 활용한다. 본 연구에서는 사람의 유일한 요소와 같은 동물의 코의 주름 형태, 땀샘 형태 등으로 이루어진 비문을 특징으로 한 생체 인식 기술을 활용하여 연구하였다.

사람의 지문인식 기술은 무려 1892년부터 경찰 수사에 본격적으로 사용될 정도로 눈부신 발전을 이룬 기술이며, 100년이 넘는 시간 동안 수많은 생체정보로써 ‘유일성, 보편성, 영속성, 그리고 획득성’을 모두 가지고 있는 혁신적인 기술로 널리 사용되어 왔다. 이러한 지문과 유사한 부분이 동물의 비문이며 동물마다 제각각인 특징을 가지고 있어 개체를 식별하는 특징으로 활용될 수 있다. 1922년 소의 비문을 최초로 연구한 이후 활발

히 연구가 이어지진 않았으나 최근의 영상처리와 AI 기술의 비약적인 발전에 힘입어 비문의 중요성이 대두되고 있다. 비문은 반려견의 특징점 정보가 다른 생체정보와 겹치지 않는, 유일하게 영구히 변화되지 않고 새로 생체정보를 등록하지 않아도 계속해서 사용할 수 있는 영속성, 다양한 환경에서 손쉽게 사용이 가능한 획득성을 가지고 있다.

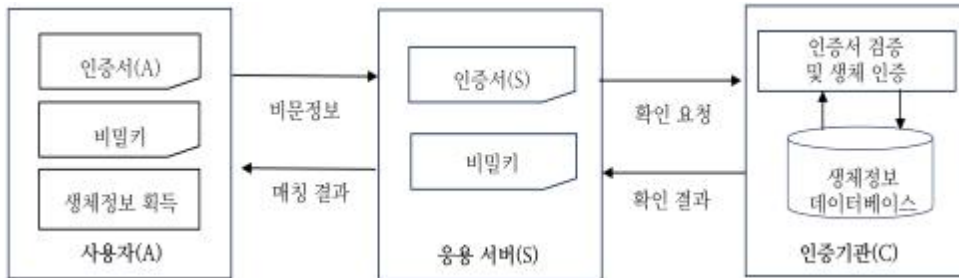
현재 국내 전체 인구의 30%가 1인 가구인 시대에 살고 있으며 반려견을 가족이자 식구로 받아들이고 있는 현실이며 이에 대한 체계적인 관리가 필요하다. 관련하여 반려동물 및 소유자에 대한 정보를 정부기관에 등록하여 동물소유자의 책임 의식을 높이며 한편으로 동물 유기를 방지하기 위한 '동물등록제'를 실시하고 있다. 동물등록은 무선전자 개체식별장치인 마이크로칩을 체내에 주사하는 내장형 삽입 방식과 목걸이 등의 형태로 부착하는 외장형 방식 중 선택할 수 있다. 내장형 방식은 한 번 체내에 삽입하면 평생 분실이나 훼손의 우려가 없는 반면, 외장형 방식은 분실·훼손 등이 우려가 있다.

반려동물의 보호와 유실·유기 방지를 위하여 반려인이 자신의 반려동물을 전국 시·군·구청에 등록하도록 동물보호법을 2014년도부터 의무적으로 시행 중이며 동물등록제는 내장형 무선식별 장치 개체 삽입 및 외장형 무선식별 장치 부착 방식으로 실행되고 있으나 실제 등록률은 38.6%로 미미한 수준이다. 이러한 반려동물의 저조한 등록률의 이유로 내장형의 경우 반인륜적으로 인식하는 부분 및 염증 발생 등에 대한 우려와 외장형의 경우 휴대 불편 등 실효성에 대한 우려가 크기 때문이다.

비문의 유일성 증명을 위해 사용되는 대표적인 평가지표로 등록 실패율(Failure To Enroll rate, FTE), 등가 오류율(Equal Error Rate, EER), 인증 실패율(Failure To Acquire rate, FTA)이 사용되고 있다(최현인 외, 2021).

통상적으로 생체인증을 이용한 PKI 시스템은 프로세스의 동작 위치에 따라 서버모델과 클라이언트모델로 구분할 수 있다. 서버모델은 [그림 2-9]와 같으며 생체정보가 데이터 수집기로부터 관련 정보를 획득하여 전

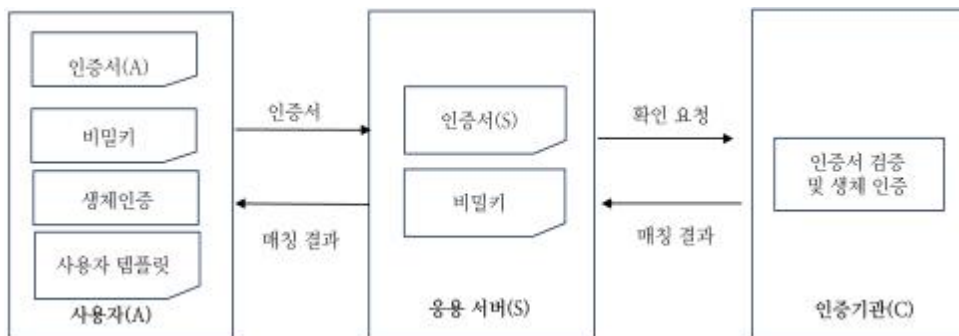
송하면 생체 정보 데이터베이스에 저장하는 방식으로 추후 생체 템플릿과 비교하여 매칭 여부에 대한 인증 결과를 전송한다.



[그림 2-9] 생체 인식 서버모델

서버모델은 관련된 생체 데이터 정보가 서버의 데이터베이스에 저장되어 있어 해킹의 위험성 및 프라이버시의 침해 문제 등이 발생할 수 있는 위험성을 가지고 있다. 따라서 생체 정보를 불법적인 접근을 방지할 수 있는 신뢰 가능한 접근 통제 메커니즘이 필요하다.

클라이언트 모델은 [그림 2-10]과 같이 수집된 데이터 입력기로부터 관련 정보를 획득하여 사용자 시스템에서 저장 되어있는 템플릿과 비교하여 데이터의 인증을 수행한다.



[그림 2-10] 생체 인식 클라이언트 모델

클라이언트모델은 데이터가 분산되어 있으므로, 클라이언트에 보관되어있는 생체 템플릿이나 프로세스에 대한 신뢰성 문제가 발생할 수 있다. 따라서, 클라이언트모델에서는 등록된 생체정보를 어떻게 안전하게 보관하느냐가 중요한 관건이다.

본 연구에서는 비문 데이터 200여개를 수집하여 Synology DS220+ NAS 시스템에 내부망을 통해 구성하였으며, VM(Virtual Machine)에 터미널로 접속하여 비문 인식 데이터를 수집 및 저장, 분석을 실행하였다. 또한 opencv-contrib-python 4.5.5.64 버전 기반에서 데이터 처리 프로세스를 위해 image Resize, CLAHE, RANSAC, DMR 과정을 거쳐 매칭을 하기 위한 원시 비문 데이터로 변환하였다. 이후 매칭 요청에 대하여 클라이언트는 요청을 서버에 보내는 것으로 트랜잭션을 개시하며 서버는 요청을 받아 해석하고, 자원들을 활용하여 응답을 클라이언트로 보내며 응답을 받은 클라이언트는 결과를 전송하는 방식으로 환경을 구성하였다.

2.3.2 비문 분석 기술 연구

비문은 동물에 있는 땀샘, 압전 등에 의하여 다양한 형태로 분포되어 나타나는 동물의 코주름을 의미하며 1900년대부터 많은 연구와 실험을 통해 사람의 지문과 같이 유일한 코드 값을 가진다는 것이 증명되었다.

대표적으로 1926년 J. G. Horning의 ‘Nose Printing Your Cat and Dog Patients’는 비문에 잉크를 바르고 종이에 찍어내는 방식으로 비문을 채취하여 비교한 최초의 비문을 이용한 개체 인식 기반 기술로 이는 지문에 잉크를 묻혀 종이에 찍어내는 지문인식 시스템에서 아이디어를 참조한 최초의 방식이다. 1922년에 W. Petersen은 350마리의 소의 비문을 조사 비교한 결과 어떤 동물들도 동일한 패턴을 가지고 있지 않다고 결론지었다. 진행한 실험은 7주령의 송아지부터 17개월령의 소까지 고루 분포한 5

마리의 소의 비문을 표본으로 5달 동안 매달 1회씩 비문패턴을 채취하였으며 소의 크기가 변하여도 비문 모양의 패턴이나, 종류가 변하지 않는 영속성(Permanent) 특징이 있음을 발견했다. 또한 Baranov et al.은 소의 비문 패턴과 품종 차이의 유전적 매개변수 추정을 다루었으며 주로 비문의 크기가 큰 소를 통한 연구를 통해 비문의 유일성을 증명하였다(Baranov, A., Graml, R., Pirchner, F., & Schmid, D., 1993).

1994년 Coldea는 개의 코를 비교 분석하여 특정한 패턴을 포함하고 있음을 연구하였으며 2002년 Minagawa는 관절 픽셀을 특징으로 주요 특징점을 추출하기 위해 이진 변환 과정과 형태학적 접근방식을 사용하여 비문의 유일성에 대한 증명을 주요 연구로 하였다. 이후 2007년 Barry 연구에서는 특징점에 대한 전처리를 통해 고유면 알고리즘에 기반한 비문 식별 방법을 연구하였으며 Minagawa 연구 방식과 유사하였으나 데이터를 세그먼트 단위로 분할하여 매칭하는 방식을 적용하여 정확도를 98.5%대로 향상하였다.

2012년 Evans and de Lahunta의 Miller's Anatomy of the Dog, 5판에서는 손가락 지문과 유사한 반려견의 코 지문이 개체를 구별하는 데 사용될 수 있음을 증명하였다.(Horning et al., 2012)

2013년 Noviyanto와 Arymurthy는 소의 비문 이미지를 이용한 특징점을 추출하기 위해 소의 코 사진을 이용하였으며 특히 후처리로 이상 값을 줄이기 위한 매칭 개선 기법을 사용하여 스케일 및 데이터의 회전성에 대한 견고함이 강조하였다. 연구에서는 비문 식별을 위해 SURF 기능을 개선한 Upright SURF(U-SURF)를 소 식별에 적용하였으며 8마리의 소로부터 마리당 15개의 이미지를 추출하여 총 120개의 비문 이미지 데이터베이스를 수집하였다. 실험 시나리오에서는 훈련 샘플에 대해 10개의 비문 이미지를 고려했고, 5개의 이미지를 테스트 샘플로 사용하였으며 회전 조건(Rotation conditions)상에서도 90%의 식별 정확도를 달성했다. Gaber et al. 는 소 식별 시스템을 개발하기 위해 AdaBoost 분류기와 결합된 비문 이

미지에서 특징 추출을 위해 Weber의 로컬 디스크립터(WLD)를 사용하였으며 획득한 최대 식별 정확도는 31마리 동물의 217개의 비문 이미지 데이터베이스를 사용하여 99%였다.

이후 인공지능 기반의 딥러닝을 적용한 연구 사례가 나타났으며 대표적으로 2017년 Kumar의 연구에서는 400마리의 소에서 소 비문을 10장씩 만들어 총 4,000개의 데이터를 기반으로 인공지능 기법 중 DBN(Deep Belief Network)을 기초로 연구하였다. 이는 기존의 연구 방식이 적용한 규칙이나 개별 프로그램 적용의 물 기반 알고리즘의 문제점인 가축의 품종이 변형되는 경우 별도의 프로그램을 적용하는 단점을 해결하였다. 그러나 DBN에서 안정적 분석을 위해서는 많은 양의 관련 이미지 데이터가 필요하다는 한계점을 가지고 있다. 이러한 한계점에도 불구하고 CNN(Convolutional Neural Network) 및 딥러닝 기법은 많은 응용 프로그램에 사용되어 물체 감지, 매칭 및 텍스트 처리 응용 프로그램 등 활발하게 사용되고 있으며 동물 생체인식 분야에 적용하여 주목을 받았다. 예를 들어, Andrew et al. 은 소 식별에 심층 학습 방법을 적용하였으며 기성 네트워크가 고정 카메라로 획득한 단편적인 하향식 이미지에서 엔드 투 엔드 방식을 적용 원거리에서 데이터의 수집 및 분석과 개인 식별을 수행할 수 있음을 보여주었다.

소의 경우 귀에 RFID 태그를 달거나 몸에 RFID 칩을 이식하는 방법이 사용하여 관리되고 있어 비문의 효율성은 개에 비해 상대적으로 낮은 형편이다. Tharwat는 범용 특징 추출 알고리즘을 사용하여 텍스처 특성 접근법에 기반한 비문 이미지 정확도를 연구하였으며 LBP(Local Binary Pattern)를 사용하여 비문 이미지에서 국소 고정 특징점을 추출하였고 식별 정확도는 99.5%를 나타내었다. Awad는 비문 인지를 위하여 SIFT(Scale Invariant Feature Transform)를 이용하였으며 연구에서는 특이치를 제거하기 위해 Random Sample Consensus(RANSAC)를 적용하여 후처리하였다. 식별 시나리오는 15마리 소에서 각 동물의 비문 이미지 7개씩을 수집

하여 총 105개의 비문 이미지를 데이터베이스에 저장하였으며 각 동물의 7개 이미지는 등록 단계와 식별 단계 사이에서 이미지 보정 하였고 105 x 105차원의 혼동 행렬(confusion matrix)을 생성한 후 RANSAC 방법으로 제안된 SIFT는 93.3 %의 식별 정확도를 달성했다.

2017년 연구된 Kumar 연구는 DBN(Deep Belief Network)을 활용한 기술로 이는 딥러닝의 일종으로 입력층과 은닉층으로 구성된 블록처럼 여러 층으로 쌓인 형태로 연결된 신경망을 활용한 방식으로 각 네트워크의 숨겨진 계층을 다음 계층의 입력으로 사용하여 학습을 통해 가중치를 근접 값으로 보정하여 마지막 튜닝과정에서 가장 가까운 최종 가중치를 계산하는 방법으로 소의 비문 인지에 사용하였다.

이는 비문 이미지의 기본 패턴을 사용하여 가축 식별을 위한 DL(Deep Learning) 기반 접근방식을 도입한 사례로 비문 포인트 이미지의 추출 된 특징을 인코딩하기 위해 잘 알려진 스택 노이즈 제거 자동 인코더 방식을 사용했다.

Zhangyong et al.는 젓소의 정확한 식별을 위해 CNN에 기반한 자동화 방법을 연구하였으며 학습 세트와 테스트 세트의 교차 검증을 통해 단일 이미지에 대한 인식 정확도가 87%에 이를 수 있음을 증명하였다.

2000년 초반부터 시작한 비문에 대한 연구와 2017년 이후 비문인식 기술은 카메라를 통해 비문의 사진을 찍은 후 AI의 딥러닝을 통한 개체 인식이 이루어지는 딥러닝 기반 연구가 진행되고 있다.

최근에 연구된 딥러닝을 활용한 반려견 비문 이미지 품질 분류 연구에서는 반려견 11마리로부터 수집된 55장의 원본 이미지를 기반으로 5,500장의 데이터 셋을 생성하여 전이학습과 미세조정을 통해 모델을 학습하고 평가한 결과, 반려견 비문 이미지 평균 품질 분류 정확도는 86.65%로 나타났다(김종복 외, 2020). 이는 비문 이미지의 학습 및 보정을 통한 이미지 왜곡을 감소시킨 것으로 비문 데이터의 정제 과정 및 데이터 확보 과정에 반영이 가능하다.

관련 연구는 특정 규칙을 인지하여 개별로 프로그래밍 후 소와 같이 큰 비문의 크기를 주로 하였으며 과거의 주요 비문 연구 사례 및 결과를 아래 [표 2-1]과 같이 정리하였다.

[표 2-1] 대표적 비문 연구 내역

년도	저자	데이터셋	주요 연구 내용	정확도
2002	Minagawa	Dog/ 43 Set	Paper-based comparison method	30%
2007	Barry	Cow/ 29 Set	Eigenvalues + Segmentation	98.5%
2013	Awad	Cow/ 15 Set	SIFT + RANSAC(Random sample consensus)	93.3%
2016	Gaver	Cow/ 31 Set	WLD + ABD	99%
2017	Kumar	Dog/ 400 Set	Deep Learning DSN Model	95.9%

최근의 딥러닝 기반 합성곱 신경망(CNN) 기능을 도입하여 인식률을 높이고 연구가 진행되고 있으며 이는 딥러닝에서 시각적 영상 분석에 주로 적용되고 있으며 영상 및 동영상 인식, 추천 시스템, 영상 분류, 의료 영상 분석 및 자연어 처리 등에 응용된다. 합성곱 신경망은 정규화된 버전의 다층 퍼셉트론으로 일반적으로 한 계층의 각 뉴런이 다음 계층의 모든 뉴런에 연결되는 신경망 구조를 나타내고 있다. CNN은 데이터에서 계층적 패턴을 활용하고 더 작고 간단한 패턴을 사용하여 더 복잡한 패턴을 표현함으로써 정규화와 같은 효과를 내는 특징을 활용 연구가 진행되고 있다.

또한 연구된 개체수가 적어 비문 인지의 특성을 충분히 반영하였다고 보기 어려우며 최적의 분석을 위해서는 관련 비문 데이터량을 많이 수집하고 생체인식 기반의 새로운 기술 적용을 통한 기존 분석 방식의 인식률 고도화가 필요하다.

2.4 연구의 차별성

현재 우리나라는 반려견을 의무적으로 등록하도록 하였으며, 이를 위해 반려견 정보 및 반려인의 인적정보 등을 담은 마이크로칩을 외장형 펜던트 또는 몸 안에 내장하는 방법 등을 사용하나 반인륜적인 문제로 인하여 저조한 등록률을 기록하고 있다.

따라서 이를 개선하기 위하여 다양한 연구를 진행하고 있으며 대표적인 연구로 바이오 인식 기술을 활용한 반려견 비문 인식 기술을 들 수 있다. 이는 사람의 지문과 같이 유일성을 내포하고 있으나 비문을 이용한 생체인식 식별자를 사용한 객체 식별 기술은 정확도 측면에서 높은 성능을 요구하고 있어 여전히 어려운 문제를 가지고 있다.

본 연구에서는 비문 관련 데이터의 정확도를 향상시키기 위하여 기존의 객체인지 알고리즘인 SURF, SIFT, BRISK 등에 대해 비교하였으며 데이터 매칭의 정확도를 향상시키기 위한 가장 많이 사용되고 있는 ORB 알고리즘을 기반으로 BEBLID의 Descriptor를 적용한 방식을 연구하였다. 실험은 반려견의 비문을 데이터베이스화하여 식별 시스템을 구축하였으며 ORB와 BEBLID 알고리즘을 이용하여 최적의 Descriptor를 도출하는 방식으로 진행하였으며 기존방식에 비해 우월함을 증명하고자 한다.

과거의 연구에서는 비문 수집의 정확성을 위해 환경의 특성상 발생하는 반사, 회전, 크기의 차이로 인한 다수의 잡음을 최소화하는데 많은 연구를 하였다. 이러한 데이터의 잡음 요소는 데이터의 인식률을 낮추는 주요 원인으로 발생 된 잡음이나 비문 이미지에서 탐지된 잘못된 데이터의 특징점을 제거하고 데이터의 정확도 향상을 위해 RANSAC 알고리즘 등을 적용하여 정확도를 높이는 방법을 사용하였다.

본 연구에서도 데이터의 정확도 향상을 위해 RANSAC, DMC 방식을 사용하였으나 이는 데이터 수집 단계의 데이터 정제 과정으로 실질적인 인식률 향상에는 한계가 있다고 보여진다.

초기 비문 데이터의 산정 시 콧구멍 사이의 주름 부분을 ROI 영역으로 선정하고 실험을 진행하였다. 선정된 ROI 영역은 굉장히 작고 세밀하기 때문에 최대한 많은 특징점을 찾아내는 것이 중요하며 이를 위하여 근사 고속 헤이시안 행렬을 사용하였다. 또한 데이터의 회전이나 사이즈의 차이에서도 정확도가 우수한 ORB 알고리즘 기반의 작은 데이터에서도 정확도가 우수한 BEBLID의 Descriptor의 값을 0~2까지 가변적으로 작용하여 최적의 값을 도출하였다.

향후 동물 등록률 제고를 위해서는 비문 기반의 생체인식 정확도 향상 기술은 절대적으로 필요하며 BEBLID 알고리즘을 활용한 적정 기술자의 값인 Descriptor를 적용하면 매칭률의 정확도가 우수할 것으로 기대된다. 등록된 비문 데이터를 효율적으로 활용되기 위해서는 비문 기반 데이터를 용이하게 수집할 수 있는 플랫폼 기반을 마련하고, 이를 수집, 가공, 저장 가능한 데이터 허브 구현으로 데이터에 대한 공유, 유통이 가능하도록 하여야 할 것이다. 특히 반려견을 키우는 반려견주의 증가로 인하여 발생하는 유기견 문제의 해결을 위해 신분증 제도 도입 및 제도 개선을 지원하는 등 새로운 융·복합 비즈니스 모델 적용이 가능할 것으로 사료된다.

비문 정보의 등록률 향상은 정확한 식별 메커니즘이 절대적으로 필요하며 이러한 기술을 기반으로 동물을 보호하고 반려견이 안전하게 관리되도록 하는 기능을 제공할 것이며, 기술적으로 반려견의 정보를 담은 QR코드와 연계한 반려견 신분증 정보의 유통 및 활용 수단을 제공함으로써, 의료, 보험, 케어 등의 다양한 분야의 관련 산업 발전에 기여하고, 유기/분실 반려견의 감소를 위한 동물 등록률을 높일 것이다.

III. 연구방법

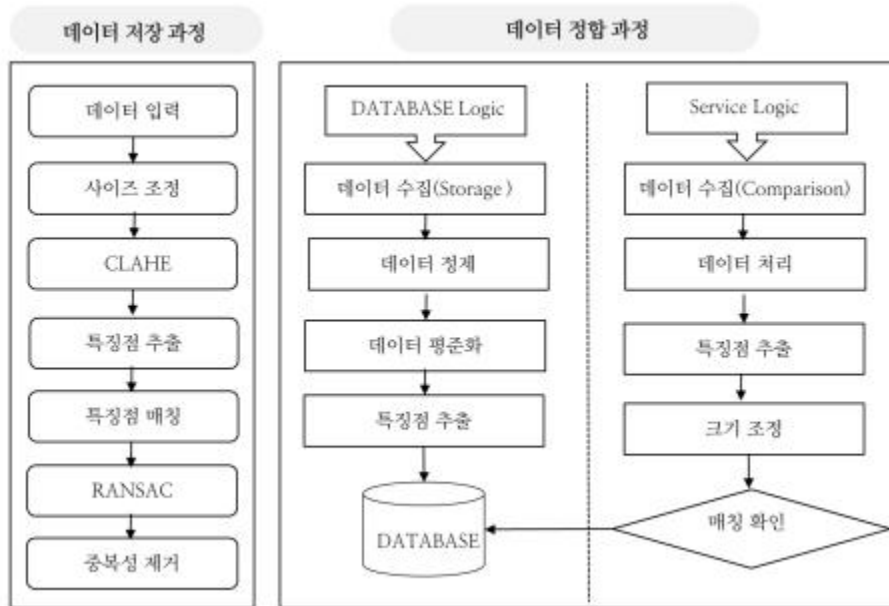
3.1 분석 방법

본 연구에서는 기존의 객체 인지 기술의 원시 데이터의 이미지 회전, 크기 변형 등의 경우 인식률 향상을 위한 방안에 대하여 실험하였으며 주로 객체 크기를 인지하기 위한 밝은 값, 키포인트 위치, 각도, 방향 등을 통해 객체 인지 및 매칭 고도화를 진행하였다. 즉 과거 연구의 주요 연구 항목은 매체 인식률 및 이미지의 정확도 향상 및 실행 시간의 단축이 주요 관심 사항이었으며 본 연구에서도 매칭률 향상을 주요 연구 과제로 추진하였다.

본 연구에서는 과거의 다양한 알고리즘에서 발전시킨 BIBLID 알고리즘의 기술자(Descriptor)의 스케일 값을 조정하여 최적의 매칭률 값을 유도하였으며 기존의 객체 인지는 차량, 인물 등 큰 대상과 관련된 객체였다면 본 연구에서는 반려동물의 작은 객체인 비문을 인지 및 매칭하는 분야에 적용하였다. 동일 비문 비교를 위해 원본 비문 데이터를 사전에 수집 후 매칭을 위한 비문 데이터의 Screening 과정을 거쳐 분석 및 정제작업을 수행하였으며 이후 비문 데이터 셋을 작성한 후 신규 수집된 비문 정보와 기존 저장된 데이터 간의 상호 비교 과정을 통해 매칭 여부를 판단하여 최종 결과를 분석하였다.

데이터 수집 단계는 비문의 모습을 감지 및 정렬하는 전처리 과정을 거쳐 객체의 특징을 추출한다. 추출된 데이터들은 데이터 허브 기반의 데이터베이스에 저장하여 비문을 보유한 원본 데이터로 데이터베이스에 축적된다. 매칭 과정은 비문을 촬영한 데이터를 원본 데이터와 비교하여 특징을 추출하고 데이터베이스에 등록 되어있는 개체인지를 판별한다. 데이터 어노테이션 과정은 비문 데이터 수집 및 선별 과정으로 모바일 카메라 어플을 통해 개의 비문 이미지를 획득하여 최대 직사각형의 관심 영역(ROI-Region of interest)을 분할 한 후 히스토그램을 사용하여 흐릿한 이미지의 해상도 보정을 거쳐 인지 가능하게 스크리닝을 수행하는 데이터 정제 과정을 수행하였다.

비문 이미지를 받아서 저장하고 이미지에서 특징점 정보를 추출 후 DB에 저장하는 API 서비스 기능을 통해 특징점의 매칭점을 실증하였다.



[그림 3-1] 비문 매칭 프로세스

데이터 수집 정제 과정은 비문 인식을 통한 동일 개체 여부 판단 프로세스 단계로, 모든 비문 이미지는 그레이스케일로 변환하여 처리하며 주어진 이미지 2개를 비교하여 matching의 개수로 동일 여부를 판단하였다. 통상 매칭점이 몇 개인지가 중요한 사항으로 Inliers 10% 이상은 매칭된 데이터로 인식되나 정확한 통계자료는 있지 않다. 매칭 하고자 하는 모델이 있을 때, 특징점 중 추정 모델의 정상 분포에 속하는 데이터는 Inlier, 정상 분포에서 벗어난 데이터를 outlier라고 특징점을 기반으로 새로운 모델을 만들고, 모든 데이터와 모델 사이의 inlier를 측정한다. Inlier가 특정 임계치 이상이면 추정 모델의 정상 분포 안에 속한다는 것을 의미한다. Descriptor는 특징점의 집합으로 특징점 주위 영역의 필요한 부분을 샘플링하여 만들며 본 연구에서는 매칭을 위해 200여 데이터 셋을 기준으로 Python3 및 OpenCV4으로 프로그램하였다.

3.2 연구 주요 내용

본 연구에서는 인지 및 매칭률 관련 다양한 알고리즘을 조사 비교 분석하였으며 이중 인지율이 높은 ORB 알고리즘에 BEBLID의 Descriptor 값을 조정하여 비문 데이터의 인지율 및 매칭 정확도 향상을 증명하는 실험을 하였다. 실험을 통해 관련 데이터를 수집 및 정제하여 매칭의 적정성을 확인하는 과정으로 데이터의 전처리, 특징 추출, 매칭 및 후처리의 네 가지 주요 단계를 거쳐 최종 데이터 매칭에 관해 실험하였다.

첫 번째 단계에서는, 이미지 크기 조정 및 히스토그램 평준화 등 부정확한 데이터를 제거하여 비문 데이터의 유효성을 향상시키는 데이터 정제단계를 수행하였다.

두 번째 단계에서는 SIFT (Scale Invariant Feature Transform), SURF (Speeded Up Robust Features), BRISK (Binary Robust Invariant Scaling Keypoints), Oriented FAST 및 ORB (Rotated BRIEF)의 주요 범용 특징 추출 알고리즘을 비교 분석하였다.

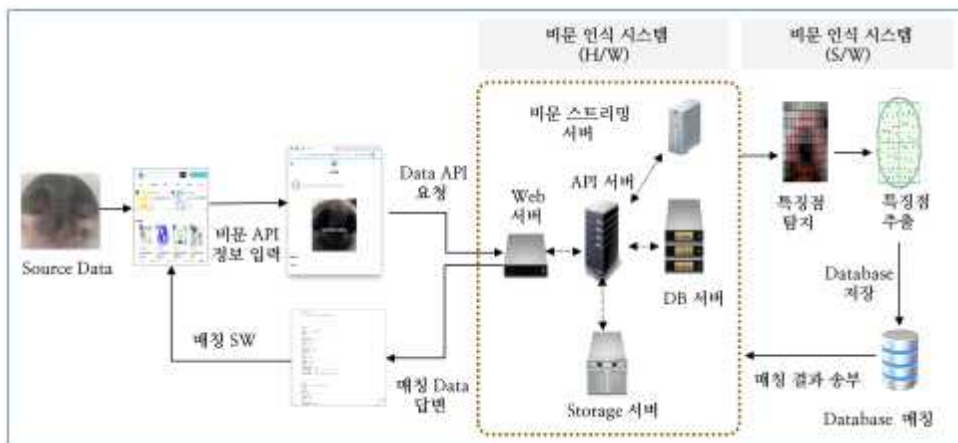
세 번째 단계에서는 인지 및 매칭의 정확도 향상을 위해 SIFT 및 SURF에 대해 FLANN(Fast Library for Approximate Nearest Neighbors) 사용 및 ORB의 스케일러 값을 조정하여 정확도를 향상하는 BEBLID 알고리즘을 적용하여 매칭 데이터의 정확성을 확인하였다.

데이터 정제과정은 216개의 원본 데이터 중 해상도를 기준으로 총 200개의 비문으로 선정하였으며 향후 입력되는 이미지의 각도, 강도, 노이즈, 사이즈 등 변형을 감안하여 최적의 인식과 매칭이 이루어지도록 하였다. 데이터의 분석 과정 이후 특징점을 추출하여 데이터베이스에 저장하였으며 특징 추출 알고리즘인 ORB와 BEBLID의 Descriptor 값을 가변적으로 적용하여 최적의 변수를 도출하였다. 이는 비문 객체에 대해 특징점 인지가 가능하여 반려견 등록률 제고 및 유기견 문제 등 사회 문제 해결을 위한 플랫폼 구축으로 확대 가능할 것이다.

3.3. 자료수집 및 연구 분석

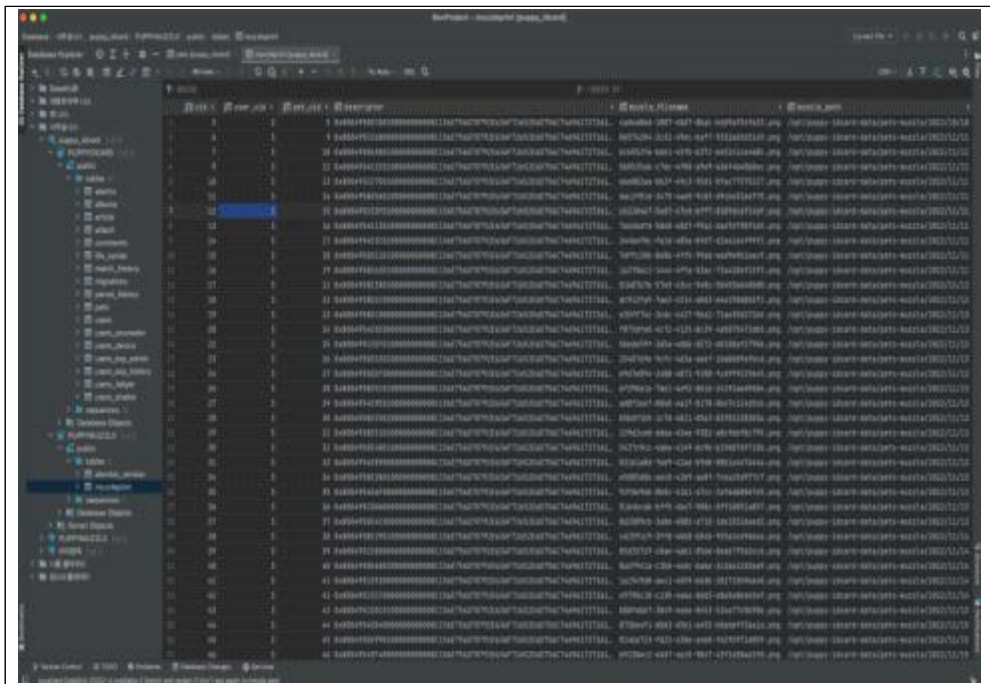
데이터 수집 정제 과정은 비문 인식을 통한 동일 개체 여부 판단 프로세스 단계로 모든 비문 이미지는 그레이스케일로 변환하여 처리하여 주어진 이미지 2개를 비교하여 matching의 개수로 동일 여부를 판단한다. 모든 비문 이미지는 그레이스케일로 변환하였으며 통상 매칭점이 4개 이상이면 동일 비문으로 인지하며 이는 경험에 의한 수치로 좀 더 정확한 학문적 접근이 필요하며 알고리즘은 Python3 및 OpenCV4를 통해 실험하였다.

비문 인지 서비스를 위한 API 및 비문 특징점 추출 및 description 저장, 매칭, 등록, 수정, 삭제 용도의 REST API 시험환경을 구성하였으며 시험대상의 성능 및 기능시험을 위한 시험환경을 구성하였다. 주요 구성 S/W로 내부 VM 생성 용도로 VMM(Virtual Machine Manager)과 설치 OS인 Cent OS7을 Python3 기반으로 구성하였다. 또한 파이썬 마이크로 웹프레임 워크 Flask와 Node Express 기반 웹프레임 워크 Nest.js 및 영상처리패키지 모듈로 Open CV-Python을 구성하였다. 기타 데이터베이스로 Postgre SQL, 데이터베이스 연결 클라이언트 툴로 Data Grip, 파이썬 개발 IDE Pycharm Professional, 웹개발용 IDE Web storm 및 컨테이너 생성 Docker로 구성하였다.



[그림 3-2] 비문 매칭 시험 환경

탑재된 시험용 S/W 용도는 펫서비스 REST API 테스트, PostgreSQL Database 접속 및 관리용도, PostgreSQL 데이터 확인으로 구성되어 있다. 데이터 모델링을 위해 10건의 시료를 비문 REST API로 등록하며 반려견의 비문 형상을 이미지 파일로 수집한 후 형상을 형태와 크기별 구분하여 데이터베이스에 저장한다. Data Screening, Image Resize, Contrast 기능을 통한 영상 데이터의 보정을 수행하며 반려동물 종별 특징점에 대한 형태와 크기를 구분하여 특징점에 대한 영상정보를 확보하였다.



[그림 3-3] 비문의 특징점 바이너리정보 저장

데이터베이스에 저장된 200여 종의 서로 다른 반려견 비교용 비문 이미지 중 불특정 반려견 비문 이미지 10종에 대하여 선정하여 비교 매칭을 위한 정확도를 시험하였다.

[표 3-1] DB 데이터 샘플 정보(10개 데이터 셋)

No	구분	반려견	비문 이미지	No	구분	반려견	비문 이미지
1	치와와 1살			6	치와와 1살		
2	치와와 1살			7	시츄 1살		
3	푸들 1살			9	치와와 1살		
4	포메라 니안 2살			9	시츄 1살		
5	치와와 1살			10	치와와 1살		

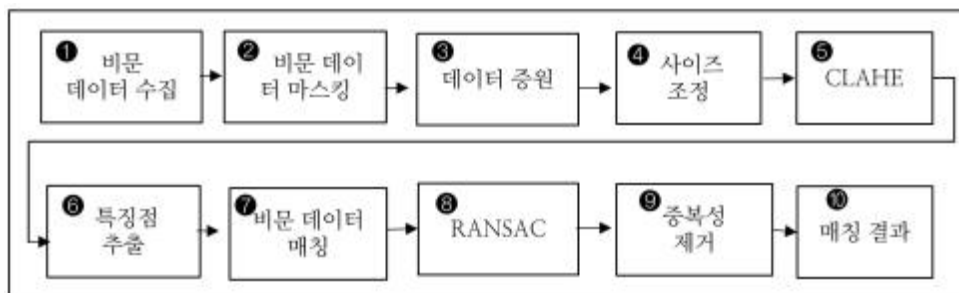
3.4 비문 분석 프로세스

3.4.1. 연구 프로세스

본 연구에서는 비문 데이터의 수집과 특징점 추출 후 분석을 통해 매칭점을 인지하는 비문 처리 10단계 프로세스를 수행하였다. 수행은 전처리와 후처리 과정을 통해 수행되며 수집된 데이터 비문과 DB에 저장된 비문 데이터의 비교를 위해 비문 데이터의 입력 ROI 영역을 설정하여 이미지를 비문 이미지로 분리하며 비문 데이터의 Screening 과정을 거쳐 전처리 작업을 수행하였다. 비문 이미지에서 추출된 특징 기술자(feature descriptors)를 사용하여 매칭이 수행하였으며 잘못된 매칭을 줄이기 위해 중복데이터의 확인 등 후처리 과정을 적용하였다.

비문 처리 과정 중 객체 인식 알고리즘을 사용하여 빛 반사, 흐림 등 불필요한 데이터를 제거하였으며 특징점을 추출하여 정합을 시행하였고 중복된 특징점 등 아웃라이어 제거 과정을 수행하였다. 작성된 비문 데이터셋은 신규 수집된 비문 정보와 기존 저장된 데이터 간의 상호 비교 과정을 통해 매칭 여부를 판단하는 최종 결과를 제공한다(Mustaqee, 2022).

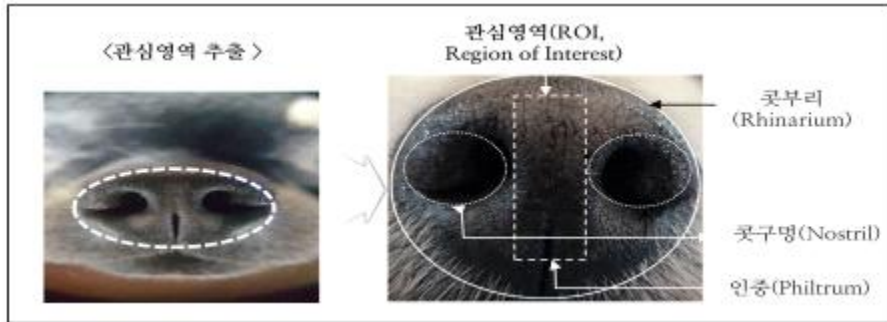
비문의 분석 및 매칭 과정은 두 객체의 비교 조건에 따라 가변성을 지니고 있으며 본 연구에서는 ORB와 BEBLID 알고리즘을 활용하여 최적의 매칭률 조건을 도출하였으며 세부 과정은 [그림 3-4]와 같다.



[그림 3-4] 비문 분석 전체 프로세스

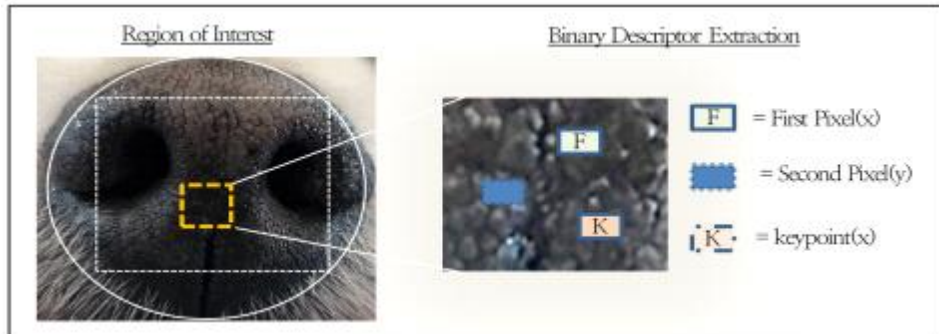
3.4.2. Dataset 구성

본 연구에서 적용한 개 비문 촬영을 위해 스마트폰 1440 x 3200 해상도에 4,608,000 화소 카메라를 사용하여 비문 이미지를 수집하였으며 촬영 중 움직임을 수반하고 있어 여러 번 촬영하여 적정 이미지를 추출하였다. 약 10cm 이내 거리에서 200마리의 개 비문 이미지에서 획득하였으며 이미지 수집은 직사광선의 빛 반사를 고려하여 실내에서 수행되었다. 이미지 촬영 후, 획득한 이미지에서 비문 식별을 위한 관심 영역 ROI를 설정하였으며 콧부리 영역(Rhinarium)에서 두 콧구멍(nostril)의 경계 지점을 포함하는 직사각형 영역으로 정의하였다. 이는 용기한 선에 따라 만들어지는 모양 또는 이 용선의 형태를 만드는 모양이 물체의 표면에 부착된 후 만들어진 자취로 변하지 않으며 유일한 형태를 지니고 있어 사람의 지문과 같은 성격을 가지고 있다(정제창, 이민정, 2017).



[그림 3-5] 비문 ROI 설정

비문 데이터는 key Point를 중심으로 Image patch 내에서 2개의 픽셀을 선정해 Descriptor를 생성하며 이는 픽셀의 분포를 결정하는 주요 factor가 된다. 첫 번째 픽셀은 Key Point를 중심으로 표준편차가 σ 인 Gaussian Distributions에서 생성하며 두 번째 픽셀은 Key Point를 중심으로 표준편차가 2σ 인 Gaussian Distributions에서 생성한다. 이러한 과정을 N번 반복해 N개의 Bit로 이루어진 Descriptor를 생성한다.



[그림 3-6] Binary Description 추출

초기 데이터 모델링 데이터 확보를 위해 시험대상의 시료를 이미지 파일로 수집한 후 형상을 형태와 크기별 구분하여 저장 Case 별로 확인하며 Data Screening, Image Resize, Contrast 기능을 통한 영상 데이터의 저장 Case 별 확인과 반려동물 종별 특징점에 대한 형태와 크기를 구분하여 특징점에 대해 영상정보 Case 별 확인을 실시하였다.

[표 3-2] 입력 데이터 테이블 구조

No	출력명	타입	데이터 (예시)
1	uid	Integer	27
2	user_uid	Integer	7
3	pet_uid	Integer	10
4	descriptor	bytea	00000000 80 04 95 0E 49 01 00 00 00 00 00 8C 15 6E 75 6DI.....num py.core.multiarr 00000020 61 79 94 8C 0C 5F 72 65 63 6F 6E 73 74 72 75 63 ay..._reconstruct
5	muzzle_name	String	b6ee7071-6054-4429-754a86a4b5.png
6	muzzle_path	String	/opt/petservice/pets-muzzle/2021/10/15
7	create_at	timestamp	2021-10-15 01:20:07.169479
8	modified_at	timestamp	null

time	statusCode	message	match	is success	imagePath
2022-12-12 00:00:00	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:00
2022-12-12 00:00:01	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:01
2022-12-12 00:00:02	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:02
2022-12-12 00:00:03	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:03
2022-12-12 00:00:04	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:04
2022-12-12 00:00:05	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:05
2022-12-12 00:00:06	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:06
2022-12-12 00:00:07	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:07
2022-12-12 00:00:08	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:08
2022-12-12 00:00:09	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:09
2022-12-12 00:00:10	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:10
2022-12-12 00:00:11	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:11
2022-12-12 00:00:12	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:12
2022-12-12 00:00:13	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:13
2022-12-12 00:00:14	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:14
2022-12-12 00:00:15	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:15
2022-12-12 00:00:16	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:16
2022-12-12 00:00:17	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:17
2022-12-12 00:00:18	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:18
2022-12-12 00:00:19	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:19
2022-12-12 00:00:20	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:20
2022-12-12 00:00:21	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:21
2022-12-12 00:00:22	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:22
2022-12-12 00:00:23	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:23
2022-12-12 00:00:24	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:24
2022-12-12 00:00:25	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:25
2022-12-12 00:00:26	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:26
2022-12-12 00:00:27	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:27
2022-12-12 00:00:28	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:28
2022-12-12 00:00:29	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:29
2022-12-12 00:00:30	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:30
2022-12-12 00:00:31	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:31
2022-12-12 00:00:32	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:32
2022-12-12 00:00:33	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:33
2022-12-12 00:00:34	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:34
2022-12-12 00:00:35	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:35
2022-12-12 00:00:36	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:36
2022-12-12 00:00:37	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:37
2022-12-12 00:00:38	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:38
2022-12-12 00:00:39	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:39
2022-12-12 00:00:40	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:40
2022-12-12 00:00:41	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:41
2022-12-12 00:00:42	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:42
2022-12-12 00:00:43	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:43
2022-12-12 00:00:44	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:44
2022-12-12 00:00:45	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:45
2022-12-12 00:00:46	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:46
2022-12-12 00:00:47	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:47
2022-12-12 00:00:48	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:48
2022-12-12 00:00:49	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:49
2022-12-12 00:00:50	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:50
2022-12-12 00:00:51	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:51
2022-12-12 00:00:52	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:52
2022-12-12 00:00:53	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:53
2022-12-12 00:00:54	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:54
2022-12-12 00:00:55	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:55
2022-12-12 00:00:56	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:56
2022-12-12 00:00:57	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:57
2022-12-12 00:00:58	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:58
2022-12-12 00:00:59	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:00:59
2022-12-12 00:01:00	1	정상적으로 등록되었습니다	1	1	2022-12-12 00:01:00

[그림 3-7] 특징 저장 데이터 셋

각각의 반려견 데이터에 대한 비문이미지는 자동으로 구분하며 각각의 비문이 등록될 때마다 정상적으로 등록되었는지에 대한 응답메세지를 확인한다. Postman에서 비문등록 API를 처리할 때 걸리는 각 단계별 소요되는 시간 정보와 Request, Response Size 정보 및 Status Code 정보를 확인한다. 이후 각각의 비문이 등록될 때 단계별 이미지인 원본이미지, resize, grayscale, contrast, keypoints 등 정보를 저장하였다.

본 연구에서는 저장후 비문 데이터베이스에 접속하여 특징점 Description 바이너리 정보와 추가 분류 부분들이 구분되어 모델링 되었는지 여부를 비문 데이터베이스에 SQL 및 DataGrip 관리툴로 확인하였다.

[표 3-3] 출력 데이터 테이블 구조

No	출력명	타입
1	statusCode	Object
2	message	Double
3	match	Double
4	isSuccess	Double

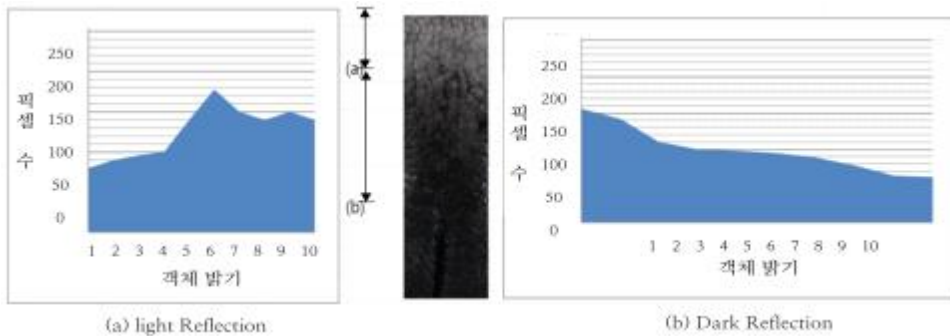
3.4.3. Data Set Making

비문 개체 식별을 위해 빛 반사가 없는 적절한 이미지와 높은 선명도를 선택하는 것이 필요하며 이는 초기 데이터 수집 시 개의 활발한 움직임으로 인해 이미지 획득이 쉽지 않기 때문이다. 이러한 문제점을 해결하기 위해 개체 인식 알고리즘을 통해 주요 특징점을 중심으로 주변의 인식률을 높이는 과정을 수행하였으며 비문 이미지에 중간값 필터(Median filter)와 언샤프 마스크(Unsharp mask) 과정을 수행하였다.



[그림 3-8] 비문 데이터 교정

초기 수집된 비문 데이터는 빛 반사로 인한 이미지의 보정이 필요하며 특히 비문의 ROI 영역은 일반적으로 어둡기 때문에 밝기값이 낮으며 빛이 반사된 영역은 밝기값이 높게 나타났다. 비문 데이터의 빛이 반사되고 빛이 반사되지 않은 이미지의 히스토그램을 [그림 3-9]에 표시하였다.



[그림 3-9] 이미지 히스토그램

선명도를 높이기 위하여 이미지 스크리닝(image screening) 적용하며 선정된 ROI 이미지에 히스토그램 이퀄라이제이션(histogram equalization)을 적용하였다. 이는 이미지의 전체적인 대비를 높이고 선명도를 올리는 효과를 나타내나 노이즈도 증폭되는 역기능도 가지고 있다(Tong, H. et al, 2004). 이러한 문제를 해결하기 위하여 AHE(Adaptive Histogram Equalization) 및 CLAHE(Contrast-Limited Adaptive Histogram Equalization)로 문제를 해결하며 이는 후처리 과정에서 수행되는 알고리즘이다.

수집된 데이터 중 알고리즘의 전체 Brightness 임계치값 50 아래의 이미지 제거와 해상도를 조정하여 각각의 이미지의 픽셀 수를 최적의 200개의 데이터 세트를 구성하였으며 ROI 이미지의 크기는 최소 290 x 280픽셀에서 최대 841 x 825픽셀까지, 평균은 549 x 515로 픽셀화하였다.

3.4.4. Data Augmentation (데이터 증대)

Data Augmentation 란 데이터의 양을 늘리기 위해 원본 데이터에 각종 변환을 적용하여 개수를 증가시키는 기법으로 일반적으로 시스템의 견고성을 증명하는 데 사용된다. 알고리즘의 견고성을 평가하고 반려견의 비문 인식을 높이기 위한 특징점 추출 알고리즘 반영의 소스 데이터는 원래 테스트 데이터 세트의 이미지 변형이 필요하다. 이미지 변형은 비문 이미지 획득 중에 발생할 수 있는 촬영 각도, 조명 강도, 원근감, 노이즈 등과 같은 요소를 고려하여야 한다.

Data augmentation은 데이터 셋을 다양한 방법으로 변형하여 실질적인 학습 데이터 셋의 규모를 키울 수 있는 방법으로 주로 이미지 데이터와 텍스트 데이터에 적용되며, 모델이 더욱 일반화되고 다양한 상황에서 작동하도록 하며 Mirroring, Random Cropping, Rotation, Shearing, Color Shifting 등의 기법을 사용한다.

데이터 증강은 모델의 성능을 향상시키고, 과적합을 줄이는 데에 도움이 된다. 또한, 실제 환경에서 모델이 잘 동작하도록 도와준다. 예를 들어, 학습 데이터에 노이즈가 없는 사진이 많지만 테스트 이미지에는 다양한 노이즈가 있는 경우, augmentation을 통해 노이즈의 분포를 예측하고 학습 데이터에 노이즈를 삽입하여 노이즈에 잘 대응할 수 있도록 한다. 데이터가 많다는 것은 데이터의 기본적인 특징을 학습하지 못할 때 발생하는 overfitting을 줄일 수 있다는 것을 의미하며 데이터 셋이 실제 상황에서 입력값과 다를 경우, augmentation을 통해 실제 입력값과 비슷한 데이터 분포를 만들 수 있음을 의미한다. Data augmentation은 데이터를 늘릴 뿐 아니라 모델이 실제로 잘 동작할 수 있도록 노이즈의 분포를 예측하고 학습 데이터에 노이즈를 삽입하여 모델이 노이즈에 잘 대응할 수 있도록 한다.

3.4.5. Image Resize

원본 테스트 데이터 세트의 크기는 반려건의 비문 식별에 있어 중요한 요소이며 이미지 크기는 최소 290 x 280픽셀에서 최대 841 x 825픽셀까지 다양하게 생산된다. 기존 Tharwat에서는 이미지의 크기를 300 x 300 픽셀, Noviyanto와 Arymurthy에서 300 x 400픽셀의 고정된 크기를 반영하였으며 당시는 소의 비문을 기준으로 하였다.

그러나 반려건의 비문은 소보다 훨씬 작기 때문에 이미지 크기를 조정하여야 하며 이때 원본 이미지의 너비와 높이 비율을 변경하면 왜곡이 발생할 수 있으며 비문 식별 성능 저하를 유발할 수 있다. 따라서 정보 손실을 최소화하기 위해 원본 이미지의 너비와 높이 비율을 최대한 유지함이 필요하다. 본 연구에서는 원본 이미지와 비교 이미지의 크기를 비율에 맞게 적절하게 조절하였으며 양선형 보간법(Bilinear interpolation)을 사용하여 크기를 조절하였다.

양선형 보간법은 원본 영상에서 인접한 실수 좌표를 둘러싸고 있는 네 개의 픽셀값에 가중치를 곱한 값들의 선형 합으로 결과 영상의 픽셀값을 구하는 방식으로 2x2 이웃 픽셀(200 x 200)로 비율에 맞게 아래 [그림 3-10]과 같이 resize 하며 비율에 맞게 변경한 비문 이미지들을 그레이스케일 이미지로 변환하였다.



[원본 비문] Grayscale



[비교 비문] Grayscale

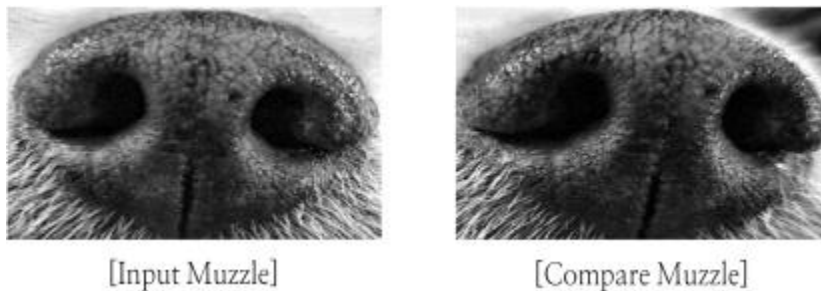
[그림 3-10] 비문 이미지 변환

3.4.6. CLAHE(Contrast Limited Adaptive Histogram Equalization)

일반적인 AHE(Adaptive Histogram Equalization)는 해당 영역의 히스토그램이 매우 집중되어 있기 때문에 영상의 거의 정수에 가까운 영역에서 대비를 과대 증폭하는 경향이 있고 결과적으로 AHE는 거의 일정한 영역에서 노이즈가 증폭될 수 있는 문제점이 있다. Contrast Limited AHE(CLAHE)는 이러한 noise amplification 문제를 줄이기 위해 대비 증폭이 제한된 adaptive histogram equalization의 변형으로 수집된 데이터의 이미지 향상을 위해 히스토그램 균등화(Histogram equalization)를 사용하며 이는 이미지에서 픽셀값의 누적 분포 함수를 사용 이중 전체 이미지의 대비(contrast)를 최적화 되게 하는 방식의 고급 히스토그램 이퀄라이제이션 방법을 사용한 것이다. CLAHE에서 주어진 픽셀값 주변의 대비 증폭은 변환 함수의 기울기에 의해 주어진다. 이는 neighbourhood cumulative distribution function(CDF)의 기울기에 비례하므로 해당 픽셀값에서의 히스토그램값에 비례한다. CLAHE는 CDF를 계산하기 전에 히스토그램을

사전 정의된 값으로 clipping하여 증폭을 제한하며 CDF의 기울기가 제한되므로 변환 함수의 기울기도 제한되게 된다. 히스토그램이 clipping되는 값(clip limit)은 히스토그램의 정규화와 인접 영역의 크기에 따라 달라진다. CLAHE는 이미지를 작은 영역으로 나누고 대비의 한계치를 정해 극단적인 노이즈를 제거하여 각 영역 내의 히스토그램을 균일화하게 하는 기술을 적용한다(Petersen, W., 1922).

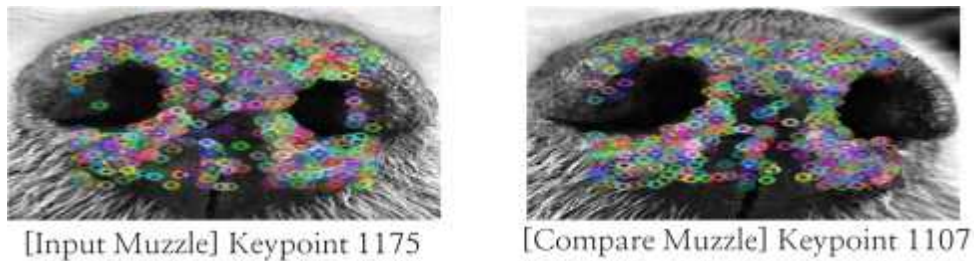
반려견의 비문 이미지의 대비(contrast)는 일반적으로 낮고 조명 조건의 차이로 인해 이미지마다 다르게 나타나며 낮은 대비와 대비의 차이는 반려동물의 비문 식별 성능을 저하시킨다. 히스토그램 평활화(CLAHE)는 비문 데이터의 어두운 부분을 밝게 하고 반대로 밝은 부분은 어둡게 하여 명암값의 분포를 균일하게 한다(Zuiderveld, K., 1994). 따라서 본 연구에서는 히스토그램이 충분히 늘어날 때까지 이미지 간의 대비 차이를 줄이고 대비를 향상시키기 위해 반복적으로 적용하였으며 CLAHE는 히스토그램에서 0-49 및 206-255에 1000픽셀 이상이 될 때까지 반복한다. [그림 3-11]은 CLAHE에 의해 강화된 이미지를 보여주며 CLAHE 구현에서 대비에 대한 임계값은 2로 설정되고 그리드 크기는 8 x 8픽셀로 설정하였다.



[그림 3-11] 블록 단위의 히스토그램 균일화(CLAHE)

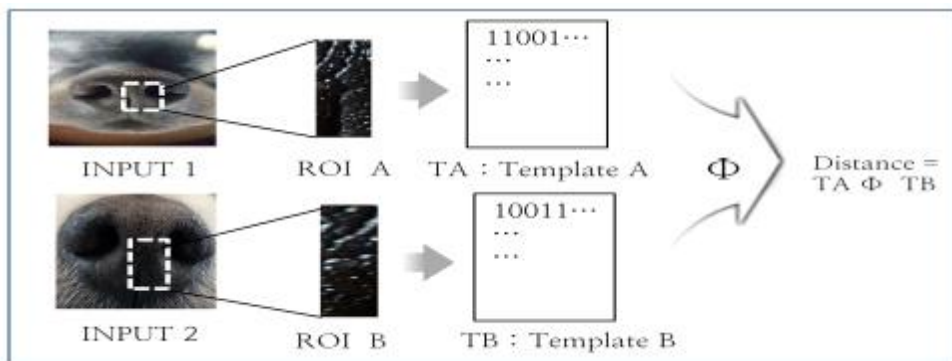
3.4.7 Feature Extraction

이미지에서 고유 특징을 추출하는 알고리즘은 키포인트 감지기 (Keypoint detector)와 Descriptor로 이루어지며 이는 특정 이미지에서 키포인트를 찾아 구분한다. 기술자는 CLAHE 처리된 비문 이미지에서 특징점 기반 해당 키포인트를 벡터 기반 설명하는 정보를 담아 저장한다.



[그림 3-12] CLAHE 처리 비문 이미지 특징점 추출

등록 비문 특징점 정보는 descriptor에 바이너리 형태로 데이터베이스에 모델링하여 저장한다. 이는 ROI 영역 내의 직사각형 어레이 위치의 값을 Gabor 변환 값 (x0, y0)의 좌표 정보로 0과 1의 값으로 부호 값으로 변환된다(최형인 외, 2021). 다음으로 해밍 거리를 사용하여 이 두 데이터의 일치성을 $d(A,B) = TA \oplus TB$ 로 나타낸다.



[그림 3-13] 비문 특징점 매칭(knn Match)

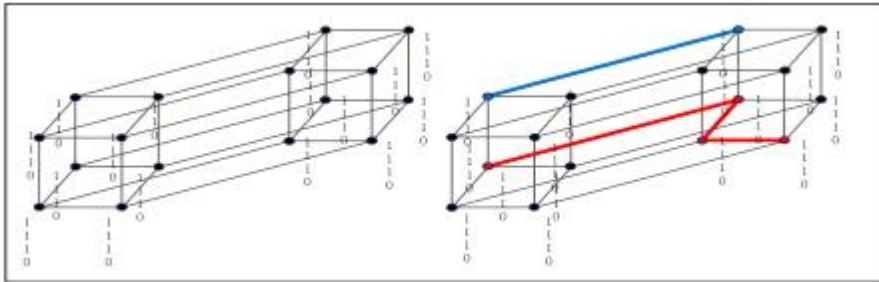
ROI A의 템플릿 TA와 ROI B의 TB를 해밍거리를 이용 추출된 특징 점 간의 거리를 통해 얼마나 유사성을 가지는지 측정하며 일반적으로, 매칭 거리가 작을수록 두 코 영상이 유사하고, 매칭 거리가 클수록, 두 코의 이미지가 다르다(한창용, 권송화, 최현인, 2018). 매칭 시험은 바이너리 특징 점 descriptor 정보를 가지고 상호매칭하여 비율을 확인하며 확인결과 특징 점 매칭 결과 506개 포인트가 도출되었으며 매칭률은 86%를 나타내었다.

3.4.8. Feature Matching

매칭 알고리즘은 거리를 계산하여 feature 추출 알고리즘에서 추출한 feature 기술자를 매칭 한 다음, 일치하는 매칭 개수를 산출한다. 매칭 일치 수는 비문 이미지가 동일한 것인지 결정하는 임계값으로 같은 길이의 두 문자열에서 같은 위치에서 서로 다른 기호의 수를 확인한다.

해밍 거리 함수 $dh(s,t)$ 는 다음과 같다.

$$d_h(s,t) = |\{i \in [0,1,\dots,n-1] \mid s_i \neq t_i\}| \quad i \in [0,1,\dots,n]$$



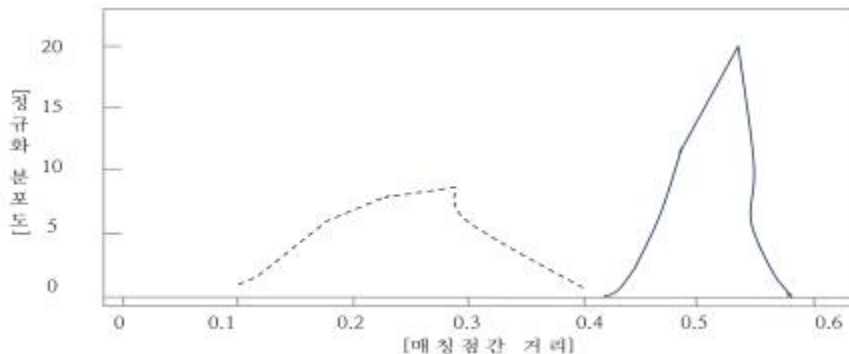
자료 출처 : <https://geo-spatial.org/vechi/articole/un-doi-i-trei-istoria-unei-margarete>

[그림 3-14] 해밍 거리 정의

4차원 정육면체에서 해밍 거리를 살펴보면 특징점 정보 0100에서 인근 특징점 정보 1001까지 거리가 3임을 알 수 있으며 특징점 0110에서 특징점 1110까지의 해밍 거리가 1임을 알 수 있다.

ORB에서 해밍 거리(Hamming distance)는 바이너리 디스크립터 매칭을 사용하였으며 해밍 거리는 이진 문자 열의 구성요소에 따라 0과 1의 값을 가지며 같을 경우 0, 다를 경우 1로 구분된다. 해밍 거리값이 작을수록 기술자 간의 오차가 적다는 것을 뜻하며 낮을수록 매칭점 후보군으로 함께 인식될 확률이 크다는 것을 의미한다.

[그림 3-15]에는 정규화된 히스토그램(확률 분포) 그래프에서 일치 거리를 비교하고 분석한 결과를 보여주며 임계값 0.4를 기준으로 분리된다는 것을 알 수 있으며 본 연구에서도 해밍 거리의 임계값은 0.4로 설정하였다(최현인 외, 2021).



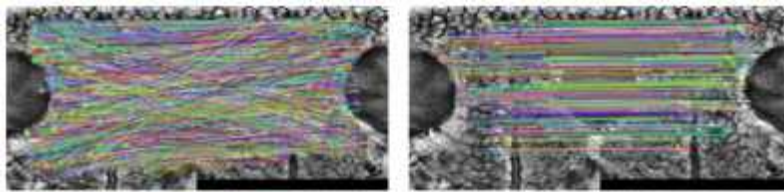
[그림 3-15] 정규화된 히스토그램(확률 분포)

특징점 추출 기술자들 간의 해밍 거리 계산은 입력 영상과 비문 데이터의 특징점 이미지간의 이루어지게 되며, 거리를 계산한 후 입력 영상과 비문 데이터 이미지 간의 매칭점 후보군들간에 매칭 방향이 동일하게 비교하여 선정하는 대칭 테스트(Symmetric Test)를 거쳐 확정하였다.

3.4.9 RANSAC(Random Sample Consensus)

최소제곱오차법(RANSAC)은 데이터 셋에서 노이즈를 제거하고 모델을 예측하는 알고리즘으로 특정 임계값 이상의 데이터를 무시하고 outlier에 강건한 특성을 가지고 있다. 특이치가 추정치 값에 영향을 미치지 않을 때 특이치를 포함하는 관측 데이터 집합에서 수학적 모델의 모수를 추정하는 반복적인 업무를 수행하여 정확도를 향상시킨다. RANSAC을 사용하여 매칭 프로세스를 수행한 후 이상치를 제거하였으며 이상 값을 포함한 관측 데이터에서 무작위로 선택된 샘플에서 수학적 모델의 매개변수를 추정한 다음 반복 작업을 통해 가장 내부값이 높은 모델을 찾는 방식으로 세밀한 비문 데이터의 최적화를 위해 적용하였다(Fischler, M. & Bolles, R., 1981).

본 연구에서는 비문의 콧구멍 사이의 인종을 포함한 영역을 ROI로 선정하였으며 특징점 좌표를 추출하여 투영 변환행렬(Projective transformation matrix)을 수행한다. RANSAC 과정은 찾아낸 투영변환 행렬을 모든 정합쌍에 적용해 가장 많은 표본이 동의한 행렬을 찾는 방식으로 [그림 3-16]에 전후 모습을 통해 상호 동의한 특징점 간의 연결도를 나타내어 준다.



(a) 아웃라인 제거 전

(b) 아웃라인 제거 후

[그림 3-16] RANSAC 알고리즘 적용 결과

RANSAC은 데이터에 포함된 outlier 비율에 관계 없이 적용할 수 있는 강력하면서도 효과적인 파라미터 추정 방법이나 동일한 입력 데이터에 대해서도 RANSAC 결과가 매번 달라질 수 있는 한계점을 가지고 있어 이를 고려하여야 한다.

3.4.10 DMR(Duplicate Matching Removal)

비문 이미지에는 유사한 특징점이 많기 때문에 데이터 매칭에서 많은 키포인트가 발생되며 이로 인해 다른 이미지의 포인트와 잘못 매칭될 수 있다. 이러한 부정확한 매칭은 잘못된 일치 횟수로 간주되며 시스템의 잘못된 일치율을 증가시킨다. 따라서 데이터의 인식률을 항상 시키기 위하여 두 개 이상의 일치 항목이 부정확할 경우 중복되는 일치 키포인트를 제거하는 DMR을 적용하였다. 일반적으로 데이터 셋 또는 데이터베이스에서 중복된 레코드 또는 엔트리를 식별하고 제거하기 위해 사용되는 프로세스 또는 시스템을 의미하며 DMR의 주요 목표는 중복되거나 동일한 정보를 제거하여 데이터의 정확성과 일관성을 보장하는 것으로 데이터의 정확성과 일관성을 보장하기 위한 데이터 관리 및 데이터 클리닝을 목적으로 한다.

데이터를 수집 및 병합하는 과정에서의 오류로 인해서 데이터가 중복되는 경우가 발생할 수 있으며 unique 한 'key' 값을 관리해야 하는 경우 중복(duplicates)이 발생하면 분석에 심각한 영향을 끼칠 수도 있다. 따라서 데이터 분석에 들어가기 전에 Seriously 중복데이터를 확인하고 처리하는 전처리 작업을 수행하며 중복이 존재하는지 확인할 때 사용할 수 있는 것이 Python pandas의 duplicated() method를 이용 처리하였다.

본 연구에서는 알고리즘과 기술을 활용하여 포맷 표준화, 결측값 처리, 데이터 종류 변환 등을 수행하여 데이터셋 내의 중복된 매칭기록을 식별하여 제거하였으며 비문 데이터의 특징점을 기준으로 상호 중복도를 점검하여 불필요하게 중복되거나 반복되는 데이터(Redundant), 데이터가 상호 겹치거나 중첩되는 데이터(Overlapping) 제거 등을 통해 최적의 데이터 셋 200개를 산정하였다.

IV. 연구 결과

4.1 비문 표본의 특성 분석

과거 연구에서는 영상의 특징점을 분석하여 비문에 대한 유일성 증명과 영상 데이터의 특징점을 수집, 분석하는 알고리즘인 SIFT, SURF, ORB 등에 대해 논하였다면 본 연구에서는 가장 많이 사용하는 ORB 알고리즘에 대하여 2022년 발표된 BEBLID에 비문 데이터를 적용하여 데이터 구성 및 매칭을 통한 인식률의 개선 사항 및 정확성 실험 결과를 도출하였다.

ORB는 2011년 Ethan Rublee 등이 처음 제시한 빠르고 강력한 국소 특징 검출기로 물체 인식 또는 3D 재구성 같은 컴퓨터 비전 작업에 사용한다. 이 기능은 FAST 키포인트 검출기와 시각적 descriptor BRIEF(이진 강건 독립적 기본 특징)의 수정된 버전을 기반으로 하며 SIFT에 대해 빠르고 효율적인 대안을 제공한다는 것이다. ORB는 영상 이미지의 로컬 기능을 감지하고 설명하는 컴퓨터 비전 알고리즘으로 객체 인식, 이미지 등록 및 비디오 추적과 같은 작업을 위해 이미지의 특징점을 정확하게 감지하고 정의한다. ORB 알고리즘은 영상의 크기가 달라지는 경우 특징점으로 검출되지 않을 수 있는 단점을 개선하였으며 여러 스케일의 영상을 이용하며 영상이 회전하였을 경우 인지율이 저하되는 문제점을 해결한 방식이다.

그러나 매체에 대한 인식률 향상과 매칭 시간의 단축이 필요하였으며 이는 이미지 일치 정확도를 높이고 실행 시간을 줄이는 새로운 기술자(Descriptor)인 BEBLID를 도출하였고 본 연구는 이를 객체인 개의 비문에 적용하여 ORB의 기술자를 개선하여 대처하는 효과를 검증하였다. 이를 위해 수집된 비문 이미지에 대해 수집 데이터의 잡음을 제거하고 구조를 명확히 하여 품질을 향상하고 적합한 특징점 정보를 추출할 수 있도록 하는 전처리 알고리즘인 Duplicate Matching Removal (DMR), 대비 제한 적용 히스토그램 평활화(CLAHE) 등 기법을 적용하였으며 BEBLID 알고리

검증 방식은 216여개의 개 비문 이미지를 수집하였으며 이중 각도, 조명, 노이즈, 변환 등을 반영한 해상도 검증을 통하여 이미지의 변형을 제외한 200개의 이미지로 데이터베이스에 저장하여 비교 데이터와의 매칭률을 평가하는 원데이터(Source data)로 설정였다.

[그림 4-1] 등록된 반려견 데이터 조회 화면

본 연구의 매칭 정확성을 위해 비문 데이터의 무결성(Data Integrity)과 데이터 샘플의 대표성은 중요한 부분으로 이를 기반으로 정보를 추출하고 비문의 정확한 매칭이 이루어진다. 데이터가 부정확하면 분석 결과에도 영

향을 미치며 매칭의 일관성이 없게 나타나기 때문에 데이터 품질과 데이터의 샘플수의 적정성은 중요한 요소이다. 특히 비문 이미지 데이터는 작은 크기로 인해 빛의 조도 및 왜곡 처리에 민감한 특징을 가지고 있어 적정한량의 정제된 데이터를 실험에 사용하여야 한다.

본 연구에서는 데이터 무결성을 위해 수집된 데이터의 해상도 조정과 이미지 사이즈 조정, 이미지 히스토그램 균일화, 특징점 추출 후 중복성 제거 등을 통한 데이터 정제 및 평준화를 이루었다.

표본의 크기를 선정함에 있어 표본의 대표성, 집단 분석 가능성, 통계검정의 정확성, 통계적 신뢰도 등의 중요 요소를 고려하여야 한다. 통계검정 수행 시 표본 크기가 작으면 결과의 불확실성이 높아지며, 표본의 크기가 클수록 검정의 정확성을 향상되나 비용과 시간이 많이 소요됨에 따라 통계적으로 신뢰할 수 있는 적정한 표본 크기의 결정이 필요하다.

본연구에서 표본 수의 결정은 초기 216개의 비문 데이터를 수집하였으며 최소 180 x 170픽셀에서 최대 841 x 825로 단계별 조정하여 평균 549 x 515으로 픽셀화된 최적화 데이터 200개를 [표 4-1]과 같이 선정하였다. 통계적 시뮬레이션 결과에 따르면, 일반적으로 최소 200개 이상의 표본이 확보되면 분석이 가능한 것으로 조사되었으며(황세웅, 2023) 비문 관련 기존의 연구에서 50개 이하의 데이터를 사용하여 시험하였음을 감안할 때 설정된 200개의 비문 데이터는 실험의 대표성을 가지고 있다고 사료된다.

[표 4-1] 해상도 기준 시험 데이터 분류

Resolution	Pixels	Training (source Data)	Test (Confirm Data)	Confirm ratio
180 x 170	30,600	216	216	100%
290 x 280	81,200	216	204	94%
454 x 430	195,220	216	200	93%

Resolution	Pixels	Training (source Data)	Test (Confirm Data)	Confirm ratio
549 x 515	282,735	216	200	93%
760 x 725	551,000	216	184	85%
841 x 825	693,825	216	160	74%

실험 설계와 연구 목적에 따라 표본의 분포가 이루어져야 하며 실험 결과의 신뢰도를 높이기 위해서는 샘플의 나이를 고려하여 [표 4-2]와 같이 적절한 표본의 크기를 결정하여 적용하였다. 또한 200개의 정제된 데이터를 정확도 기반하에 분석이 올바르게 진행되지 않으면 신뢰도가 떨어지기 때문에 opencv-contrib -python 4.5.5.64 버전 기반에서 프로그램을 개발하여 데이터를 분석하였다.

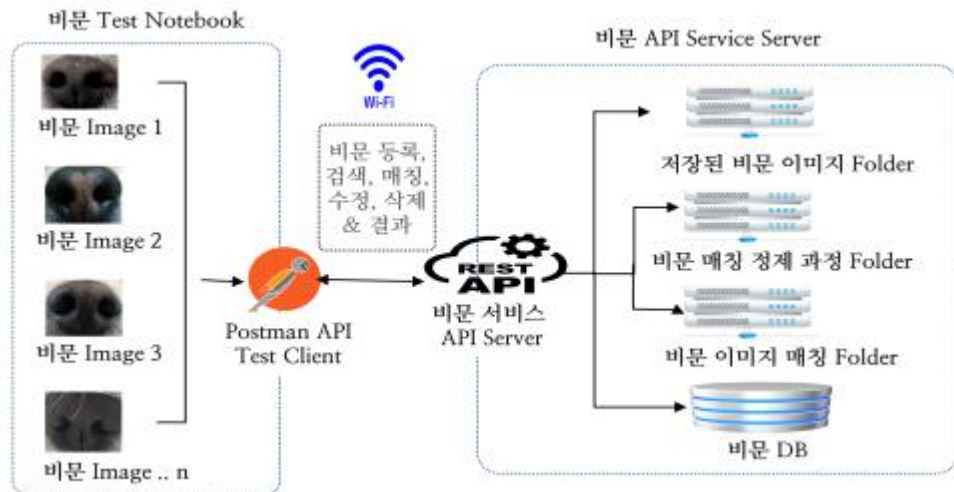
[표 4-2] 확정 시험 데이터 분류

나이	최초 데이터	확정 데이터	확정 비율
3세 이하	43	40	20%
4~ 6세	27	25	13%
7~9세	45	45	23%
10~12세	46	40	20%
13~ 15세	37	35	18%
16세 이상	18	15	8%
계	216	200	100%

4.2 실험 결과

4.2.1 비문 인식률 시험 절차

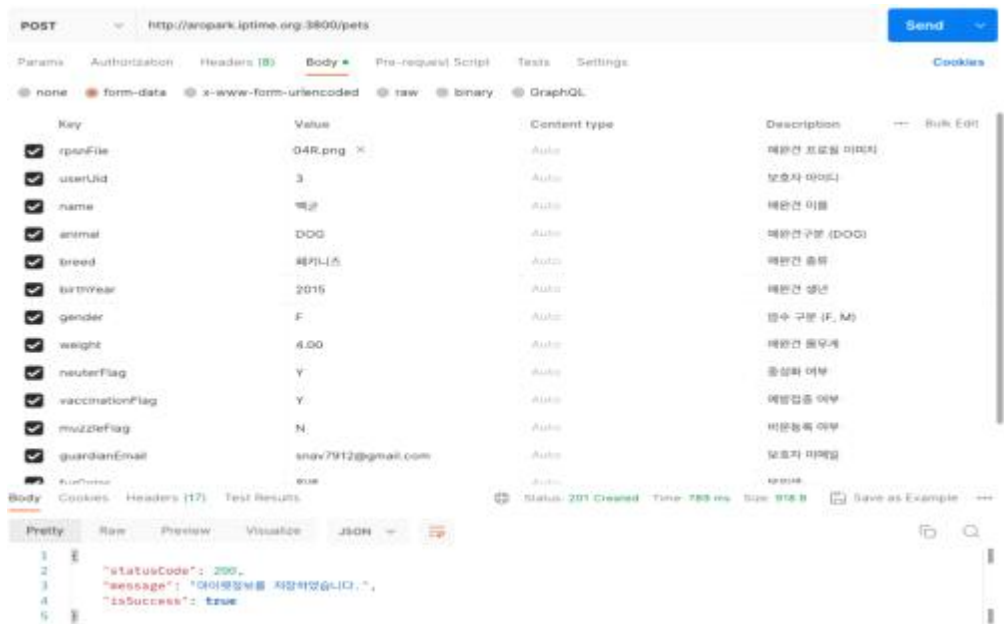
비문 인식 REST API 서버에 등록 되어있는 반려견의 비교를 위한 매칭용 이미지를 촬영하여 파일로 준비하며 비문 인식 REST API 서버에 등록되지 않은 반려견의 비문을 매칭시켜 보기 위하여 등록되지 않은 반려견 비문 이미지도 준비하였다. 시험 단말에서 Postman을 실행한 후 비문 매칭 REST API를 호출하고 기등록 되어있는 각각의 반려견의 또 다른 비문 이미지를 첨부하여 매칭 여부를 요청하였으며 각각의 비문 매칭 요청에 대한 성공 및 실패의 응답 메시지를 확인하였다.



[그림 4-2] 비문 매칭 흐름도

Postman에서 비문 매칭 API를 처리할 때 걸리는 각 단계별 소요되는 시간정보와 Request, Response Size 정보 및 Status Code 정보를 확인한다. 매칭이 성공하였을 경우 응답 메시지에 원본, 비교 비문 이미지의 특징점 개수와 전체 매칭점과 정상 매칭점, 그리고 정상치 비율을 확인하였으며 정상치 비율이 10% 이상이면 동일한 반려견으로 판단하였다.

비문 매칭 결과가 이미지로 저장되는 저장 폴더에서 원본, 비교 비문 이미지와 각각의 특징점 표시 및 정상 매칭점에 대한 연결 선분을 확인한다. Postman에서 비문 매칭 REST API를 호출하고 비문 이미지로 등록되지 않은 다른 반려견의 비문 이미지를 올려 각각 매칭 여부를 요청하며 반려견의 비문 이미지로 반복해서 시험하며 등록된 데이터 개수를 조회하고 저장된 데이터셋을 [그림4-3]과 같이 최종적으로 확인하였다.

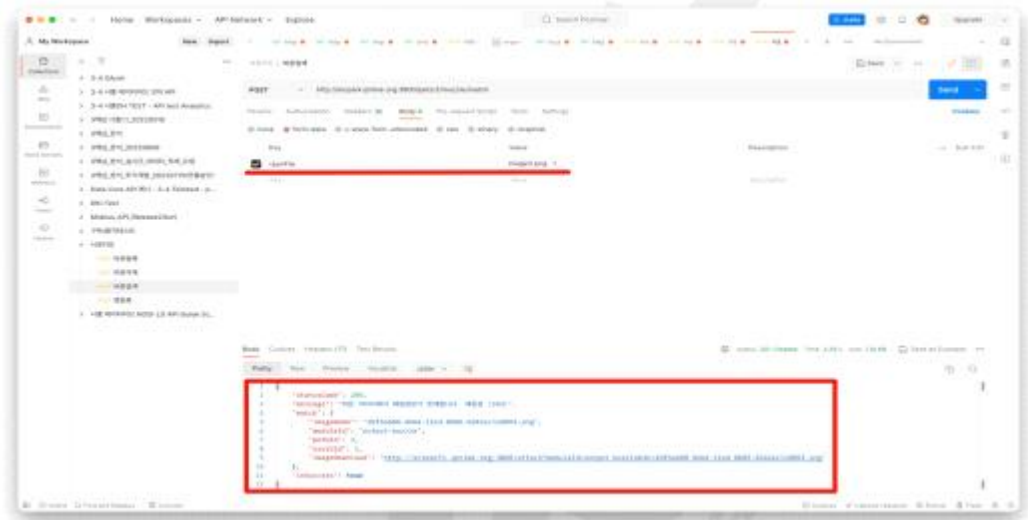


[그림 4-3] 비문 데이터 등록 정보

비문 정보는 반려견의 비문 이미지에서 크게 변화에 영향을 받지 않는 특징점(Key points)들을 찾아내고, 각각의 특징점 주변 픽셀 정보에 대한 기울기 방향 및 크기 등, 여러 정보를 함께 설명한 해 놓은 특징 디스크립터(Descriptor)를 추출하여 저장하게 되는데 이 정보는 하나의 특징점에 대한 정보들이고 하나의 비문 이미지에서 이러한 특징점을 최대 10,000개 까지 추출할 수 있다.

본 연구에서는 실제 반려견의 이미지의 특징점들을 설명해 놓은 특징

디스크립터의 각각 정보들을 하나하나 매칭해서 확인하면서 전체 유사도를 체크하는 해밍 거리 알고리즘에 의한 근접한 거리 매칭 방식을 적용하였다.



[그림 4-4] 비문 데이터 검색

현재 매칭의 유사도는 Cross Check 방식으로 두 개의 특징 디스크립터 중 하나에서 매칭을 찾은 결과와 반대의 경우에서도 찾은 결과가 있다면 매칭으로 인정한다. 매칭된 결과에도 잘못 매칭되는 경우가 있으므로 RANSAC 알고리즘을 이용하여 확인하며 정상 매칭점을 인식하여 매칭 비율을 계산한다. 정상치 비율(정상 매칭점/ 모든 매칭점), 모든 매칭점, 정상 매칭점을 화면에 표시하여 특징 디스크립터의 매칭 부분의 비율을 알 수 있으며 테스트 결과 동일한 반려건의 비문 이미지를 가지고 매칭 확인한 결과 촬영방향이 다른 각도이고 정상적인 방향에서 촬영되지 않아도 40% 이상의 매칭 비율이 나타났으며 다른 반려건의 비문 이미지일 경우 비율이 10% 이하의 매칭률이 나타나는 것으로 조사되었다. 본 연구에서는 10% 이하의 매칭률이 나오면 동일한 반려건이 아닌 것으로 판단하였다.

반려건 비문 데이터셋 200건에 대해 비문 코드 등록을 수행하였으며 비문 데이터 검색 및 매칭을 통하여 확보 매칭을 통한 인식 여부를 시험

하였다. 이를 위해 비문 이미지를 저장하고 이미지에서 특징점 정보를 추출 후 DB에 저장하는 API 서비스 기능 제공 등 인지와 매칭을 하기 위한 주요 API 서비스 기능 등을 구현하였다. 비문 데이터 플랫폼에서는 등록된 비문 이미지와 특징점 정보를 수정하는 API 서비스 기능과 전체 등록된 비문 목록을 보여주는 API 서비스 기능, 특정 비문 이미지에서 특징점 정보를 추출 후 비문 DB에 저장된 비문과 매칭 여부를 확인할 수 있는 API 서비스 기능 제공 및 비문 이미지 관리 및 매칭된 이미지를 관리하는 기능 등으로 구성되어 정확도 및 관리 기능을 구현하였다.



[그림 4-5] 비문 매칭 결과

저장된 데이터베이스의 비교 데이터셋은 비문 서비스 Data 모듈과 함께 상호 작용하며 반려견 비문에 대한 API 서비스를 하는 백엔드 서비스, PET-ID API와 PET-MUZZLE API로 각각 서비스 구성하였으며 데이터의 비교 시 저장된 데이터베이스에 있는 기존데이터와 10% 이상의 매칭률을 가지는 경우 동일 데이터로 인지토록 하였다. 매칭을 검증한 결과 100%의 매칭 및 인식 성공률을 나타내었다.

4.2.2 DB 데이터와 인터페이스 기술

DB 데이터와 인터페이스 연결 매칭 시간을 확인할 수 있도록 비문 매칭이 가능한 반려건의 비문 이미지를 준비하며 NAS의 VM(Virtual Machine)에 터미널로 접속하여 비문 인식REST API 서버와 펫서비스 API 서버를 구동한다. 시험용 단말에서 DataGrip를 실행시키고 비문 데이터베이스에 접속하고 Postman S/W를 실행하였다.

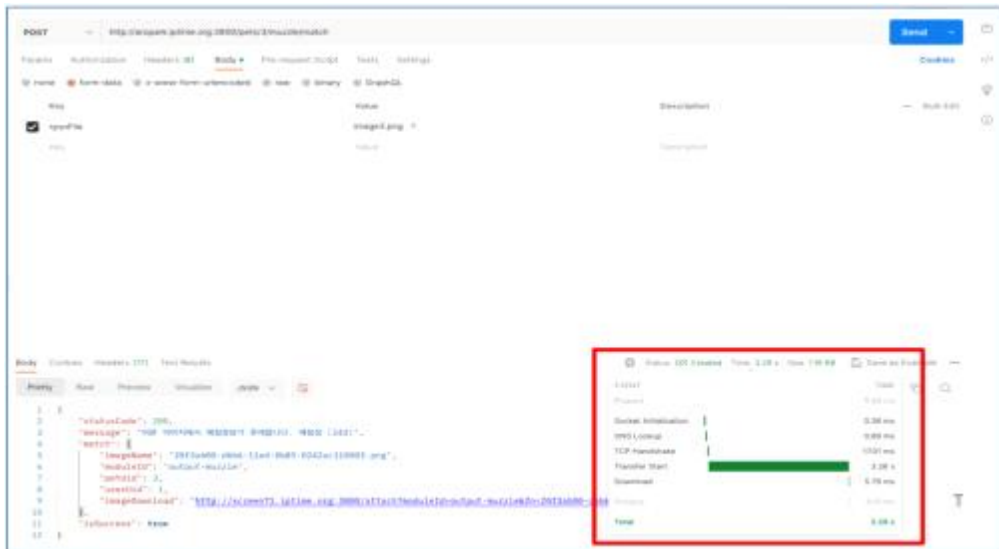
이후 시험용 단말에서 Postman을 실행한 후 비문 매칭 REST API를 호출하고 기등록된 각각의 반려건의 또다른 비문 이미지를 첨부하여 매칭 여부를 요청하며 각각의 비문 매칭 요청에 대한 성공 및 실패의 응답 메시지를 확인한다. Postman에서 비문 매칭 API를 처리할 때 걸리는 Time 정보인 통합된 시간을 확인하였다.

이후 데이터베이스에 저장 되어있는 비문에 대한 데이터 정보와 Postman에서 비문 매칭 API 호출 시 Time 정보를 기준으로 연계 시간을 확인한다. 이는 인터페이스 연계 부분의 경우 웹API 요청부터 시작해서 DB 접속 및 데이터 Query, Matching 그리고 결과를 응답으로 전송하고 받는 것까지 최종시간을 테스트하므로 API 요청부터 최종응답을 받을 때까지 소요되는 통합된 시간을 확인하였다.

REST API HTTP 프로토콜 기반으로 서버에 관련 API를 요청하고 요청된 API에서 Database를 접속 및 검색하고 관련 데이터를 매칭하기 위한 서버모듈의 작업, 그리고 최종서버 작업 후 그 결과를 다시 HTTP 프로토콜을 이용하여 응답하고 최종적으로 이용자가 응답된 결과를 다운받으며 전체 인터페이스 연계 시간을 알기 위해서 API 요청과 최종 결과를 다운받은 후 걸리는 시간을 확인한다. Postman으로 비문 매칭 API를 호출하고 매칭 결과를 받을 때까지의 Prepare, Socket Initialization, DNS Lookup, TCP Handshake, Transfer Start, Download, Process의 과정으로 각각 확인하였다.

최종 제안된 방법의 처리 시간은 Intel Celeron J4125 쿼드코어 2.0 GHz 및 64.0GB RAM 하드웨어 구성으로 측정되었다. 처리 시간은 특징 추출 알고리즘에 의해 감지된 키포인트 수에 따라 달라지므로 이미지마다 다르다.

DB 데이터와 인터페이스 연결 매칭 시간을 확인할 수 있도록 비문 매칭이 가능한 반려견의 비문 이미지를 카메라를 통해 10건을 준비하였다. NAS의 VM(Virtual Machine)에 접속하여 비문 인식 REST API 서버와 펫서비스 API 서버를 띄우고 관련 소프트웨어를 실행시키고 비문 데이터베이스에 접속하고 매칭을 확인하였다. REST API HTTP 프로토콜 기반으로 서버에 관련 API를 요청하고 요청된 API에서 Database를 접속 및 검색하고 관련 데이터를 매칭하기 위한 서버 모듈의 작업, 그리고 최종서버 작업 후 그 결과를 다시 HTTP 프로토콜을 이용하여 응답하고 최종적으로 이용자가 응답된 결과를 다운받을 때까지 과정을 거치게 된다.



[그림 4-6] 인터페이스 연계 시간

전체 인터페이스 연계 시간을 알기 위해서는 API 요청과 최종 결과를 다운받은 후 걸리는 시간을 확인하여 연계 시간을 측정하였다.

Postman으로 비문 매칭 API를 호출하고 매칭 결과를 받을 때까지의 최종시간을 보여주는 부분이 있으므로 그 부분을 확인하면 통합시간을 알 수가 있다. Prepare, Socket Initialization, DNS Lookup, TCP Handshake, Transfer Start, Download, Process의 과정으로 각각의 시간을 측정하며 확인된 최종시간은 [표 4-3]과 같다.

[표 4-3] 비문 데이터 매칭 시간

수행 번호	비문 등록여부	조회 소요시간(초)	시험결과
1	등록 비문	10.99	• 등록 비문 10.56초 (10,560 ms)
2	등록 비문	10.26	
3	등록 비문	10.95	
4	등록 비문	10.17	
5	등록 비문	10.45	
평균		10.56	
1	미등록 비문	36.42	• 미등록 비문 37.97초 (37,970 ms)
2	미등록 비문	35.09	
3	미등록 비문	39.09	
4	미등록 비문	38.91	
5	미등록 비문	40.37	
평균		37.97	

동일한 비문을 비교한 경우 200개의 데이터 셋을 기준으로 평균 10.56초 소요 되었으며 미등록 비문과 조회한 결과 37.97초로 3배 이상 소요되었다. 이는 데이터베이스에 저장되어 있지 않는 비문을 조회, 매칭한 결과로 Inliner가 10% 미만일 경우 미등록으로 설정하였으며 100% 등록 비문을 조회 매칭 확인되었다.

4.3. 적용 프로그램 함수 분석

ORB 알고리즘에 BEBLID Descriptor 값 적용은 비문 데이터의 정확도 향상을 위하여 특징 매칭(Feature Matching)을 사용한다, 특징 매칭은 서로 다른 두 이미지에서 특징점과 특징 디스크립터들을 비교해서 비슷한 객체를 연결 비교하는 의미한다. 특징 매칭을 위해 Open CV의 특징 매칭 인터페이스 함수를 사용하여 이미지 inliers1 및 inliers2 변수 내에서 일치 항목 및 일치 시 결과를 확인한다. OpenCV 3.4에서 제공하는 특징 인터페이스 함수 매칭기로 BFMatcher와 FLannBasedMatcher 등이 있으며 cv2.DescriptorMatcher_create() 함수를 이용하였다. 생성된 특징 매칭기는 두 개의 디스크립터를 서로 비교하여 매칭 해주는 함수로 match(), knnMatch(), radiusMatch()를 사용한다. 세 함수 모두 첫 번째 파라미터인 queryDescriptors를 기준으로 두 번째 파라미터인 trainDescriptors에 맞는 매칭을 찾는다.

첫 번째 함수 matches는 매칭 결과 및 DMatch 객체의 리스트를 나타내며 matches: matcher.match(queryDescriptors, trainDescriptors, mask)를 응용하여 1개의 최적 매칭점을 찾는다. 파라미터 queryDescriptors는 특징 디스크립터 배열, 매칭의 기준이 될 디스크립터, 파라미터 trainDescriptors는 특징 디스크립터 배열, 매칭의 대상이 될 디스크립터, 파라미터 mask(optional)는 매칭 진행 여부를 의미한다. match() 함수는 queryDescriptors 한 개당 최적의 매칭을 이루는 trainDescriptors를 찾아 결과를 나타낸다. 그러나 최적 매칭을 찾지 못하는 경우 반환되는 매칭 결과 개수가 queryDescriptors의 개수보다 적을 수 있다. matches = matcher.knnMatch(queryDescriptors, trainDescriptors, k, mask, compactResult)로 파라미터 compactResult는 매칭이 없는 경우 매칭 결과에 불포함(default=False)되며 K개의 가장 근접한 매칭을 나타낸다. knnMatch() 함수는 queryDescriptors 한 개당 k개의 최근접 이웃 개수만큼 trainDescriptors

에서 찾아 반환한다. k개의 최근접 이웃 개수는 가장 비슷한 k개만큼의 매칭 값을 반환한다는 의미이다.

`matches = matcher.radiusMatch(queryDescriptors, trainDescriptors, maxDistance, mask, compactResult)`는 `maxDistance` 이내의 거리 매칭을 표시한다. `radiusMatch()` 함수는 `queryDescriptors`에서 `maxDistance` 이내에 있는 `trainDescriptors`를 찾아 반환한다. `CompactResult`에 default값이 `False`가 전달되며 `True`가 전달되면 아무것도 추가하지 않는다.

세 함수인 `match()`, `knnMatch()`, `radiusMatch()` 함수의 반환 결과는 `DMatch` 객체 리스트이며 `queryIdx`와 `trainIdx`로 두 이미지의 어느 지점이 서로 매칭되었는지 정보를 제공한다. 또한 `distance`로 거리 인지 정보를 제공한다. OpenCV에서 `cv.drawMatches`는 매칭 결과를 시각적으로 표현하기 위해 두 이미지를 하나로 합쳐서 매칭점끼리 선으로 연결하는 작업을 `cv2.drawMatches(img1, kp1, img2, kp2, matches, flags)`을 사용한다. 파라미터 `img1`, `kp1`는 `queryDescriptor`의 이미지와 특징점을 `img2`, `kp2`는 `trainDescriptor`의 이미지와 특징점을 나타내며 `matches`는 매칭 결과와 `flags`는 매칭점 그리기 옵션으로 결과 이미지를 생성한다.



[그림 4-7] API 기반 비문 데이터셋 매칭 확인

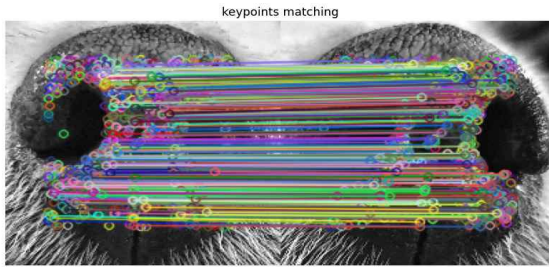
거리 측정 알고리즘을 전달하는 파라미터로 ORB로 디스크립터 검출
 기의 경우 NORM_HAMMING을 주로 사용한다. 각 해당 키 포인트는
 Perspective-n-Point, 호모그래피 추정(Homography estimation) 이미지 스
 티칭(images stitching), 평면 추적(plane tracking), 실시간 포즈 추정
 (Real-time pose estimation)과 같은 이미지 매칭 작업을 수행하며 이를
 ORB를 적용한 매칭 정확도를 측정한 결과는 다음과 같다.



[그림 4-8] ORB 사용 정상 매칭률

분석 결과 매칭을 하기 위해 촬영한 원본 데이터(Data1)의 특징점
 1,164개와 DB에 저장된 비교 데이터(Data 2) 이미지 특징점은 1,164개로
 나타났으며 이를 매칭 하였을 경우 413개의 매칭 가능한 특징점 중 276개
 가 매칭되는 것으로 나타나 67%의 매칭률을 나타내었다.

ORB와 BEBLID를 접목한 알고리즘에서 Descriptor의 스케일값(K)을
 0.25~1로 가변적으로 적용하였으며 본 매칭률은 K값을 1로 하여 매칭률
 을 실험하였다.



```

Run
/Users/ben/PycharmProjects/pythonProject/venv/bin/python /Users/ben/PycharmProjects/pythonProject/main.py
Matching Results
*****
# 원본비문이미지 특징점 : 1164
# 비교비문이미지 특징점 : 1105
# 모티브매칭점 : 442
# 정상매칭점 : 337
# 정상치비율 : 0.76%

Process finished with exit code 0

```

[그림 4-9] BEBLID 사용 정상 매칭률

ORB와 BEBLID를 비교한 결과 [표 4-4]와 같이 ORB 보다 BEBLID를 이용했을 때 정상 매칭 비율이 전반적으로 더욱 높게 나타남을 알 수 있다. 대표적으로 기술자의 신뢰성을 가장 잘 반영하는 정량적 평가 메트릭 방식인 인라이어의 백분율을 다음과 같이 계산하여 반영하였다.

$$Inliers\ Matching\ Rate = 100 * \frac{No.\ of\ Inliers}{No.\ of\ Matches}$$

[표 4-4] ORB와 Descriptor 함수 비교

구분	Keypoint 1	Keypoint 2	Matches	Inliers	Percent of Inliers
ORB	1164	1105	413	276	66.8%
BEBLID	1164	1105	442	337	76.2%

BEBLID 기술자를 사용하면 76.24%의 인라이어를 얻을 수 있으며 ORB 기술자의 인라이어 결과가 66.82%로 보다 우수함을 알 수 있다.

결론적으로 ORB와 BEBLID 알고리즘을 비교할 때 두 이미지의 일치 결과를 BEBLID에서 평균 10% 향상시킬 수 있음을 나타내며 이는 비문의 일치에 큰 영향을 미칠 수 있음을 알 수 있다. 또한 BEBLID는 이진 문자열인 0과 1을 사용하여 이미지의 작은 부분 특징점을 분석하며 설명하며 다음 이미지의 동일한 부분을 감지할 때 두 이진 문자열이 매우 유사하게 인지하며 저장한다. 이를 통해 BEBLID는 하나의 이미지를 설명하는 데 2 밀리초 미만으로 더 빠르고 정확하게 인지한다. 반려견 비문 데이터셋 200 건에 대해 실험을 수행하였으며 비문 데이터 검색 및 매칭을 통하여 일치 여부를 시험하였다. 이를 위해 비문 이미지를 저장하고 이미지에서 특징점 정보를 추출 후 DB에 저장하는 등 인지와 매칭을 하기 위한 주요 API 서비스 기능을 구현하였다.

본 연구의 데이터 Value-up은 원본 비문 데이터의 해상도 검증을 통한 카메라로 획득한 원시 영상 데이터에 대한 전처리 후 원본 비문 데이터를 정제하는 과정을 수행하였다. 주요 API로 등록된 비문 목록을 보여주는 API 기능, 등록된 비문 이미지의 특징점 정보를 수정하는 API 기능과 특정 비문 이미지에서 특징점 정보를 추출 후 비문 DB에 저장된 데이터와의 매칭 여부를 확인할 수 있는 API 기능 및 매칭된 이미지를 관리하는 API 기능 등으로 구성되어 있으며 데이터의 비문 정확도 및 관리 기능을 구현하였다.



[그림 4-10] Data API 기반 데이터 서치

4.4 EVELID 알고리즘의 적정 Descriptor 값 산정

4.4.1 특징 추출 알고리즘

동일한 비문 이미지 2개를 가지고 테스트를 진행하였으며 기본적인 작업환경은 opencv-contrib-python 4.5.5.64 버전 기반에서 테스트하였다. 이미지 처리 프로세싱 및 매칭은 기본적인 부분에서 처리되어야 할 Image Resize, CLAHE, RANSAC, DMR 과정으로 처리하였다.

특징점 추출은 SIFT, SURF, BRISK, ORB 알고리즘을 활용하였으며 각 특징점 별로 기본으로 같이 제공되는 Descriptor와, 별도로 BEBLID Descriptor의 Scaler factor 0.25~2까지 매칭률을 조사하였다.



[그림 4-11] 비문 매칭 데이터 비교

BEBLID의 Descriptor scale factor는 이미지 피라미드의 스케일링 요소를 의미한다. 이미지 피라미드는 동일한 이미지의 다양한 해상도 버전을 포함하는데, 각 수준은 원본 이미지의 다른 스케일이다. scale_factor는 두 연속적인 피라미드 수준 사이의 상대적인 스케일을 결정한다. scale_factor가 2.0이면 각 피라미드 수준은 이전 수준의 반 정도의 해상도를 갖게 된다.

본 연구에서는 2 이상의 값은 비교되는 크기와 해상도의 차로 인한 오류 데이터가 발생하여 반영하지 않았으며 이미지 피라미드의 스케일링

요소를 조절하면, 특성 검출과 매칭에 대한 성능과 속도를 다르게 조절할 수 있다. 일반적으로 더 낮은 스케일 요소는 더 정확한 결과를 가져올 수 있지만, 계산 비용이 더 많이 들 수 있어 적정값의 도출이 중요하다.

4.4.2 BEBLID Descriptor 값 실증

[표 4-5]는 SIFT, SURF, BRISK, ORB의 네 가지 특징 추출 알고리즘에 BEBLID의 Descriptor 값을 가변적으로 변화하며 실험한 매칭의 정확도 결과를 보여준다. 특징 추출 알고리즘은 공간에서 이미지 스케일 공간을 생성하여 키포인트의 최대 특징점을 찾는데 이때 추출한 키포인트의 Descriptor는 기울기의 크기와 Local Neighborhood의 상대적인 방향 계산을 통해 최적값을 계산한다. 관련하여 SIFT, SURF, BRISK 알고리즘 기반에서 BEBLID Descriptor 값을 변화하여 측정한 결과 SIFT는 10% 이하, SURF, BRISK는 60% 이하의 정합도를 나타남을 알 수 있었다. 즉 스케일 값의 변경에 대해서는 BEBLID 사용 시에도 주변 인지율이 상대적으로 떨어지고 매칭률이 낮아짐을 알 수 있었다.

[표 4-5] 알고리즘 기반 BEBLID 값 적용 실증

구분	저장 정보	입력 정보	Descr-i ptor	Scaler factor	Matches (RANSAC + DMR)	Inliers	Percent of Inliers
SIFT	414	302	SIFT	none	149	64	43%
			BEBLID	0.25	67	7	10%
				0.5	61	6	10%
				0.6	65	6	9%
				0.65	76	6	8%
				0.75	82	7	9%
				0.8	77	6	8%
				0.85	72	7	10%
				0.9	72	7	10%

구분	저장 정보	입력 정보	Descr-i ptor	Scaler factor	Matches (RANSAC + DMR)	Inliers	Percent of Inliers
				0.95	72	7	10%
				1	79	7	9%
				1.5	79	6	8%
				2	84	8	10%
SURF	343	430	SURF	none	217	78	36%
			BEBLID	0.25	62	12	19%
				0.5	115	35	30%
				0.6	119	53	45%
				0.65	117	48	41%
				0.75	112	53	47%
				0.8	113	54	48%
				0.85	117	64	55%
				0.9	121	64	53%
				0.95	121	62	51%
				1	128	63	49%
				1.5	147	76	52%
				2	144	78	54%
BRISK	873	1116	BRISK	none	315	180	57%
			BEBLID	0.25	258	11	4%
				0.5	274	14	5%
				0.6	257	29	11%
				0.65	262	27	10%
				0.75	288	48	17%
				0.8	285	55	19%
				0.85	280	61	22%
				0.9	303	70	23%
				0.95	284	73	26%
				1	287	84	29%
				1.5	254	117	46%
				2	295	166	56%
ORB	1150	1498	ORB	none	480	386	80%
			BRIEF	none	395	304	77%
			BEBLID	0.25	397	141	36%
				0.5	465	308	66%
				0.6	509	354	70%
				0.65	510	386	76%
				0.75	515	451	87%
				0.8	513	381	74%

구분	저장 정보	입력 정보	Descr-i ptor	Scaler factor	Matches (RANSAC + DMR)	Inliers	Percent of Inliers
				0.85	507	413	81%
				0.9	517	420	81%
				0.95	515	411	80%
				1	510	364	76%
				1.5	537	399	74%
				2	552	354	64%

실험은 OpenCV 4.5.5.64 Version에서 200개의 비문 데이터를 대상으로 진행하였으며 입력 이미지와 저장된 이미지 간의 매칭률을 조사하여 동일 객체 여부를 판별하였다.

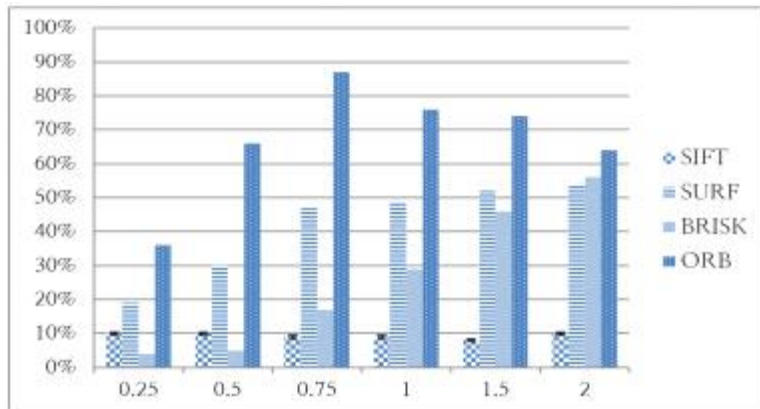
[표 4-6]에서 알고리즘별 BEBLID Descriptor 값을 0.25에서 2까지 가변적으로 적용하여 실험한 결과 평균적으로 0.75에서 87%로 가장 높게 나타남을 알 수 있다.

[표 4-6] 가변 Descriptor 값에 따른 알고리즘별 매칭률 비교

값 알고리즘	0.25	0.5	0.6	0.65	0.75	0.8
SIFT	10%	10%	9%	8%	9%	8%
SURF	19%	30%	45%	41%	47%	48%
BRISK	4%	5%	11%	10%	17%	19%
ORB	36%	66%	70%	76%	87%	74%

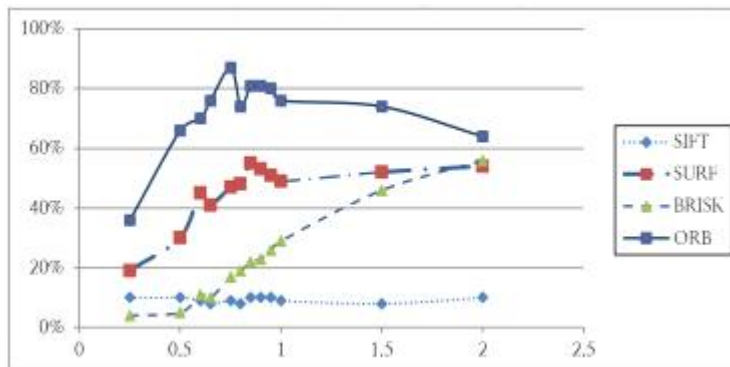
값 알고리즘	0.85	0.9	0.95	1	1.5	2
SIFT	10%	10%	10%	9%	8%	10%
SURF	55%	53%	51%	49%	52%	54%
BRISK	22%	23%	26%	29%	46%	56%
ORB	81%	81%	80%	76%	74%	64%

알고리즘별 BEBLID 가변 Descriptor 값 비교 결과 ORB에서 가변값 0.75에서 가장 매칭률이 높게 나타났으며 BRISK 알고리즘을 적용한 결과 2에서 가장 높게 나타났다. 이는 이미지 데이터의 크기에 따라 매칭률의 차이가 나타남을 알 수 있었다.



[그림 4-12] Descriptor별 매칭률 그래프

[그림 4-13]의 가로축은 BEBLID Descriptor 값의 변화를 나타내며 세로축은 매칭률을 나타낸다. Inliner가 일치하지 않을 경우 0으로 나타나며 ORB 알고리즘에서 매칭률이 높게 나타남으며 0.75대에서 높은 매칭률이 나타났다.

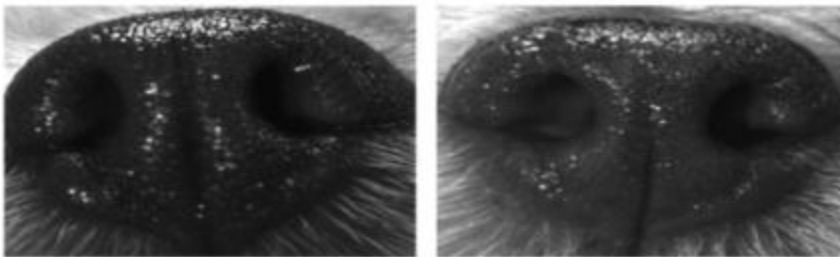


[그림 4-13] 알고리즘별 매칭률 그래프

실증 결과 각 특징점 추출 알고리즘에서 제공하는 Descriptor를 같이 사용하는 것이 단독으로 하나의 알고리즘보다 서로 잘 어우러져서 매칭률이 높게 나타났다. SIFT 나 SURF의 경우 특징점 알고리즘 자체가 실수 체계 기준으로 특징점이 구성되다 보니 ORB, BRISK처럼 바이너리 기반으로 계산하는 부분으로 매칭률의 변화가 크게 차이가 나타나지 않았다.

바이너리 기반의 특징점 추출에서 BEBLID Descriptor 값을 조정하면 매칭률의 상승을 가져왔으며 특히 ORB 을 이용하여 특징점을 추출하고 BEBLID로 Descriptor를 계산하는 부분에서는 가장 높게 나타났음을 증명하였다. 즉 ORB로 특징점 추출을 하고 BEBLID로 Descriptor 계산을 하는 것이 가장 매칭률 성능이 좋은 것을 알 수 있다.

본 연구에서는 OpenCV 릴리스 4.5.1를 이용하여 ORB 키포인트의 스케일을 다양하게 적용하여 ORB의 기술자를 개선하여 대처하는 효과를 비문에 적용하여 실험하였다. BEBLID는 매우 효율적인 설명자를 활용하며 비문의 스마트폰 카메라의 이미지 파일을 CPU에서 이미지 당 2000개의 디스크립터를 추출하여 빠르게 매칭점을 체크한다. 실행 시간을 줄이면서 이미지 매칭 정확도를 높일 수 있는 새로운 BEBLID Descriptor를 활용하여 이미지 매칭을 확인한다,



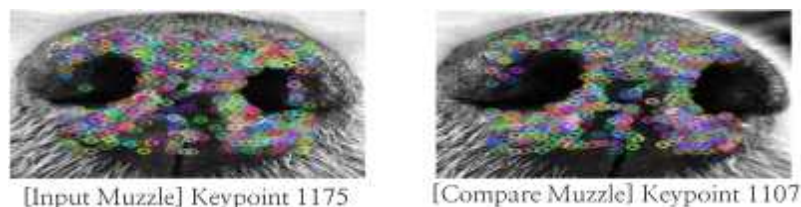
[그림 4-14] BEBLID 기반 이미지 매칭 실험

[그림 4-14]는 동일한 개의 비문이며 우측은 데이터베이스에 저장된 원본 데이터이며 좌측은 데이터 매칭을 위해 촬영한 데이터로 두 이미지 매칭을 실행하고자 한다. 이미지 매칭을 위해서는 두 이미지 사이에 정확

한 기하학적 변환이 필요하다. 이를 위해 두 이미지 데이터를 homography 형태의 3x3 행렬로 분할 하며 첫 번째 이미지에서 균질 좌표를 곱하면 두 번째 이미지의 점 좌표가 로드된다.

다음 단계는 이미지의 일부 부분을 탐지하는 것으로 먼저 ORB를 사용하여 빠르고 신뢰할 수 있는 탐지기를 사용하여 모서리를 탐지한다. ORB는 서로 다른 척도에서 모서리를 비교하고 FAST 또는 Harris 응답을 사용하여 최적의 모서리를 탐지한다. 이후 로컬 패치 1차 모멘트를 사용하여 각 모서리 방향을 찾는다. 각 이미지에서 최대 10,000개의 모서리를 탐지한다. 다음으로 로컬이미지의 기능을 감지하도록 다른 이미지에서 쉽게 찾을 수 있도록 이미지 기능을 모니터링하며 초기에 빠르고 신뢰할 수 있도록 ORB 알고리즘을 활용하여 모서리를 감지한다.

ORB는 서로 다른 스케일에서 비교하고 FAST 또는 Harris 응답을 사용하여 최상의 모서리를 선택하기 위해 특징점이 강한 모서리를 감지하며 이때 로컬 패치 10,000차 모멘트를 사용하여 너무 많은 특징점 추출로 인한 소요 시간 등을 줄이며 각 모서리의 방향을 찾도록 한다. BEBLID Descriptor은 특징 포인트의 수에 따라 이미지, 감지 정확도에 영향을 미치므로 최대한 많이 추출해야 한다(Zibo X. & Dongyu P., 2023). 비문 이미지에서 최대 500개 이상의 특징점 모서리를 감지할 수 있도록 설정하였으며 [그림 4-15]에서 점으로 표시된 특징점을 볼 수 있다.



[그림 4-15] Keypoint 이미지 판별

위의 그림에서처럼 서로 다른 이미지에서 주요 특징점을 찾아 매칭하기 위해서는 각 키포인트를 선정하여야 하며 이는 특징점 모서리 주변에

있는 이미지에 대한 숫자 벡터로 표현되며 이때 BEBLID에서는 ORB 키포인트의 스케일을 0.25~2로 설정하여 시험한다. 인수값을 2로 지정할 경우, 이미지 크기가 절반이 되는 고전적인 이미지 피라미드를 의미한다.

다음으로 두 비교 이미지의 키포인트 매칭 관계를 확인하기 위하여 첫 번째 이미지의 각 기술자와 두 번째 이미지의 모든 기술자를 비교한다. 이때 기술자 비교는 해밍거리기법을 사용하여 각 기술자 간의 다른 비트 수를 계산하여 일치 여부를 확인한다. 또한 기술자 간의 유사도율을 검사하는 비율 테스트(ratio-test)를 함께 하여 기술자 1과 2가 서로 비슷한 경우 누락되는 경유가 발생되지 않도록 한다. 이때 이미지 2의 특징점과 이미지 1에 투사된 이미지 2의 점이 2.5픽셀 미만인 경우 유효한 것으로 간주하여 기하학적 변환을 통하여 일치 항목이 올바른지 확인할 수 있도록 한다.

4.5 데이터 매칭률 조사

4.5.1 동일 데이터 매칭

인식 프로세스 단계는 비문 데이터 촬영한 후 신호처리를 기반으로 전처리 과정을 거쳐, 비문 특징점을 추출하고 데이터베이스에 저장하며 이후 비교를 위한 원본 비교 비문 데이터의 특징점과 기존에 데이터베이스에 저장되어 있는 비문의 특징점들을 비교 분석하여 인식률을 조사하였다.

```
def keypoint_process(img, name):
    tempimg = img

    # ORB & BEBLID process
    kp_detector = cv2.ORB_create(10000) # ORB 생성
    keypoint = kp_detector.detect(tempimg, None) # ORB 로 Key point 추출
    # desc_detector = cv2.ORB_create() # ORB
    desc_detector = cv2.xfeatures2d.BEBLID_create(0.75) # BEBLID 디스크립터 생성 (scale_factor 값을 75%로 축소)
    .. descriptor = desc_detector.compute(tempimg, keypoint) # BEBLID 디스크립터 계산하여 descriptor 에 결과를 넣음.
    return keypoint, descriptor
```

[그림 4-16] 비문의 특징점들 비교 분석 프로그램

keypoint_process(img, name) function으로 묶어놓은 상태로 필요에 따라 파라미터에 원본 이미지와 비교 이미지의 keypoint와 descriptor를 리턴해 주며 리턴 된 descriptor를 가지고 특징 매칭을 하게 된다. 해당 function은 아래와 같이 사용되는데 사용 전에 미리 이미지 로드하고 resize 및 그레이스케일 처리하고 CLAHE 전처리까지 하게 되면 원본이미지와 매칭이미지의 객체를 function에 넣어주면 해당 이미지의 keypoint와 특징 descriptor를 받을 수 있다. 아래는 해당 특징 디스크립터로 매칭을 시켜보는 부분의 코드이다. BEDBLID를 사용할 때 scale_factor 값을 넣어줌에 따라 인식률이 크게 달라지는데 이 값은 특징 디스크립터를 만들 때 하나의 이미지를 축소비율이 다른 여러 개의 이미지로 나열해서 이미지 피라미드를 만든 다음 해당 이미지 피라미드에서도 동일하게 특징이 나타나는 부분을 특징 디스크립터로 만들기 위한 기본 설정값이다.

```
matcher = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
matches = matcher.match(descriptorSrc, descriptorMat)
# 매칭 결과를 거리기준 오름차순으로 정렬
matches = sorted(matches, key=lambda x: x.distance)
# 매칭점으로 원본 변환 및 영역 표시 -
src_pts = np.float32([ keypointSrc[m.queryIdx].pt for m in matches ])
dst_pts = np.float32([ keypointMat[m.trainIdx].pt for m in matches ])
# RANSAC으로 변환 영역 근사 계산
mtx, mask = cv2.findHomography(src_pts, dst_pts, cv2.RANSAC, 5.0)
matchesMask = mask.ravel().tolist()
res2 = cv2.drawMatches(resizeSrcIag, keypointSrc, resizeMatIag, keypointMat, matches, None,
                        matchesMask = matchesMask,
                        flags=cv2.DRAW_MATCHES_FLAGS_NOT_DRAW_SINGLE_POINTS)

accuracy=float(mask.sum()) / mask.size
print('Matching Results')
print('*****')
print('# 원본비율이미지 특징점 : ', len(keypointSrc))
print('# 비교비율이미지 특징점 : ', len(keypointMat))
print('# 모든매칭점(Matches) : %d'%mask.size)
print('# 결상매칭점(Inliers) : %d'%mask.sum())print('# 결상치비율(Percent of Inliers) : %.2f'%accuracy)
```

[그림 4-17] 특징 디스크립터 매칭 프로그램

Max 값이 1이고 이 부분은 이미지 피라미드를 만들 때 축소하지 않고 같은 사이즈의 이미지 피라미드를 만든다는 것으로 만약 scale_factor 값이 0.75면 이미지 피라미드의 값을 각 단계별로 75% 축소해서 나열하게 됨을

의미한다. 예를 들어 처음 이미지가 1000x1000이면 다음 나열된 이미지는 75% 축소해서 750x750으로 나열하고 다음 레벨의 이미지에서는 이전 750x750에서 75% 축소해서 563x563으로 축소된 픽셀로 이미지 피라미드를 구성하게 된다. 즉, 각 단계별로 scale_factor 값으로 축소하며 각각 축소된 이미지 피라미드를 겹쳐서도 동일한 특징 부분을 보이게 되는 것을 특징 디스크립터로 생성하게 되는데 이렇게 함으로써 색상, 회전, 스케일 등에도 전혀 영향을 받지 않고 해당 이미지의 특징을 찾아서 매칭 할 수가 있게 한다.

DB에 저장된 총 200개의 이미지 데이터와 촬영된 원본 비교 이미지의 매칭률 확인을 위해 각기 동일한 객체 데이터 여부에 대한 인식을 10회 실험하였다. opencv-contrib-python 4.5.5.64 버전과 python 3.6 기반에서 BEBLID의 Descrpor의 계수값 K는 0.75로 하였다. 객체 인지 여부를 확인한 매칭 결과 평균 86.2%로 나타났으며 비문과 같은 작은 객체의 측면에서 높은 정확도를 나타내었다.

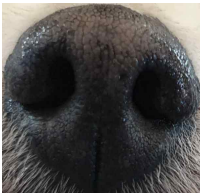
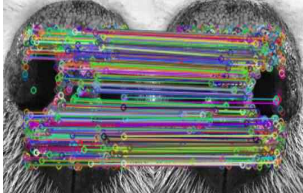
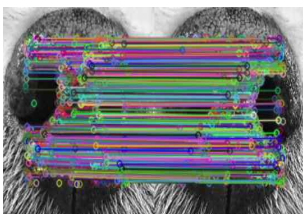
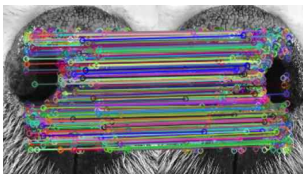
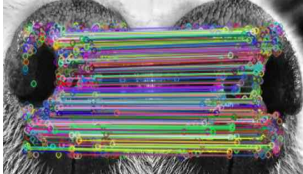
[표 4-7] 동일 비문 매칭 결과

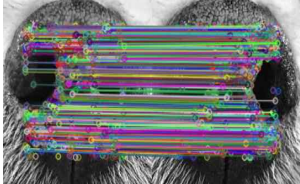
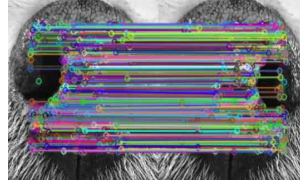
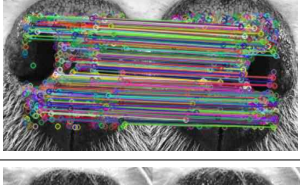
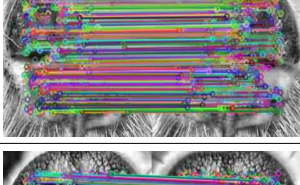
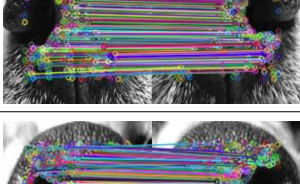
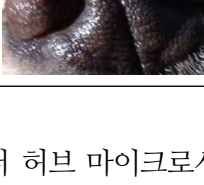
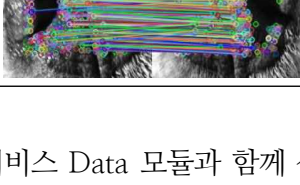
구 분	원본이미지 특장점	비교이미지 특장점	모든 매칭점	정상 매칭점	정상치 비율
Case 1	1,164	1,150	523	446	85%
Case 2	1,164	1,283	727	702	97%
Case 3	1,164	1,498	615	530	86%
Case 4	1,164	1,105	442	337	76%
Case 5	1,164	1,241	582	547	94%
Case 6	1,164	1,132	827	815	99%
Case 7	1,164	1,150	493	397	81%
Case 8	1,706	1,611	782	639	91%
Case 9	1,355	1,117	556	444	80%
Case 10	1,570	1,525	638	465	73%

본 연구에서는 비문 데이터의 특징점 추출 알고리즘인 ORB에 BEBLID의 Descriptor 값을 가변적으로 조정하여 특징점 간 거리를 계산하여 추출한

feature 설명자를 매칭 한 다음, 매칭 개수를 출력하였다. 매칭 일치 수는 비문 이미지가 동일한 비문에서 나온 것 인지의 여부를 결정하는 값으로 10%의 임계값(Threshold)을 설정하였으며 이는 임의로 설정된 값으로 바이너리 디스트립터 매칭에 사용하여 좀 더 정확한 값을 도출할 필요가 있다. BEBLID의 Descriptor 값 K를 0.75로 설정한 ORB 알고리즘에서 동일 비문에 대한 이미지 매칭 실험 결과는 아래 표와 같다.

[표 4-8] 동일 비문 이미지 매칭 비율

구분	비교이미지	매칭이미지	매칭 비율
1			<pre>***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 1150 # 모든매칭점 : 523 # 정상매칭점 : 444 # 정상치비율 : 0.85%</pre>
2			<pre>***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 1283 # 모든매칭점 : 727 # 정상매칭점 : 702 # 정상치비율 : 0.97%</pre>
3			<pre>***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 1498 # 모든매칭점 : 615 # 정상매칭점 : 530 # 정상치비율 : 0.86%</pre>
4			<pre>***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 1105 # 모든매칭점 : 442 # 정상매칭점 : 337 # 정상치비율 : 0.76%</pre>

구분	비교이미지	매칭이미지	매칭 비율
5			***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 1241 # 모든매칭점 : 582 # 정상매칭점 : 547 # 정상치비율 : 0.94%
6			***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 1132 # 모든매칭점 : 827 # 정상매칭점 : 811 # 정상치비율 : 0.99%
7			***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 1150 # 모든매칭점 : 493 # 정상매칭점 : 397 # 정상치비율 : 0.81%
8			***** # 원본비문이미지 특징점 : 1706 # 비교비문이미지 특징점 : 1611 # 모든매칭점 : 702 # 정상매칭점 : 639 # 정상치비율 : 0.91%
9			***** # 원본비문이미지 특징점 : 1355 # 비교비문이미지 특징점 : 1117 # 모든매칭점 : 556 # 정상매칭점 : 444 # 정상치비율 : 0.80%
10			***** # 원본비문이미지 특징점 : 1570 # 비교비문이미지 특징점 : 1525 # 모든매칭점 : 638 # 정상매칭점 : 461 # 정상치비율 : 0.73%

데이터 허브 마이크로서비스 Data 모듈과 함께 상호 작용하며 비문에 대한 API 서비스를 하는 백엔드 서비스, PET-IDCARD API와 PET-MUZZLE API로 각각 서비스 구성하여 매칭을 시도한 결과 100% 나타났다.

4.5.2 비동일 비문 매칭

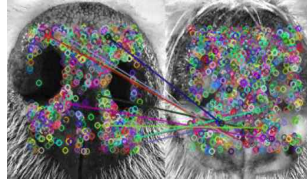
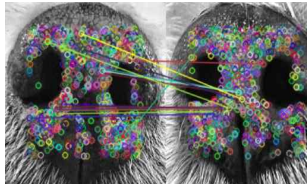
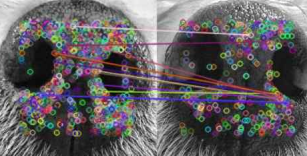
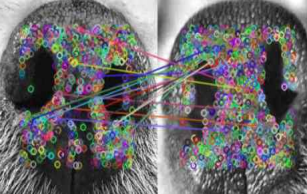
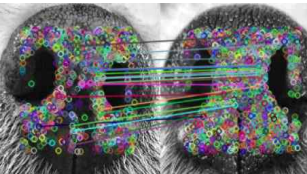
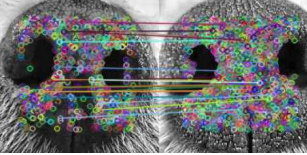
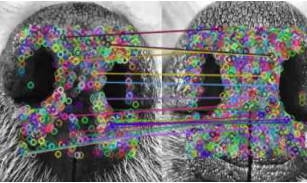
데이터의 비교 시 데이터베이스에 저장 되어있는 기존데이터와 신규 데이터 간의 매칭률을 조사하여 객체의 인지 여부를 실험하였다. 총 200개의 저장된 이미지 데이터와 저장되어 있지 않는 신규 촬영된 이미지의 매칭률 확인을 위해 각기 다른 종류 10종을 비교 데이터를 카메라로 신규로 촬영하였으며 비문으로 opencv-contrib-python 4.5.5.64 버전과 python 3.6 기반에서 다른 객체 인지 여부를 확인한 결과 다음과 실험 결과가 나타났다.

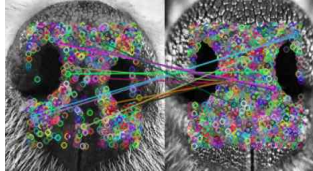
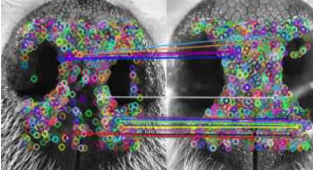
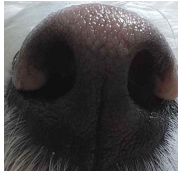
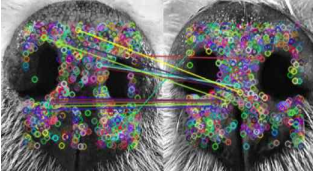
[표 4-9] 비동일 비문 매칭 결과

구 분	원본이미지 특장점	비교이미지 특장점	모든 매칭점	정상 매칭점	정상치 비율
Case 1	1,164	1,708	285	12	4%
Case 2	1,164	1,116	272	14	5%
Case 3	1,164	429	187	12	6%
Case 4	1,164	1,716	343	16	5%
Case 5	1,164	2,001	400	22	6%
Case 6	1,164	2,428	393	19	5%
Case 7	1,164	2,422	392	18	5%
Case 8	1,164	2,608	393	14	4%
Case 9	1,164	1,701	343	17	5%
Case 10	1,164	2,001	400	22	6%

실험 결과 평균 5.1로 낮은 정확도를 나타냈으며 이는 설정된 Threshold 값 10% 이하로 상호 일치하지 않는 데이터임을 알 수 있었다.

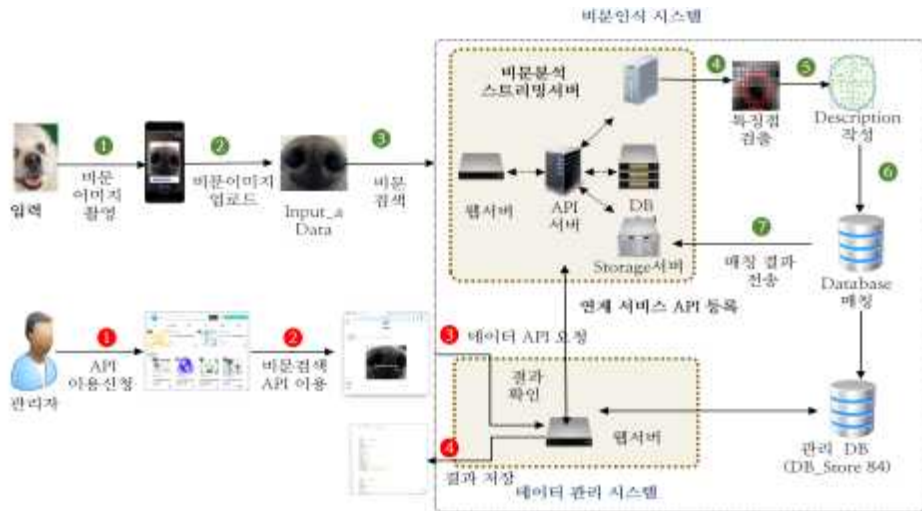
[표 4-10] 비동일 비문 이미지 매칭 비율

구분	비교 이미지	매칭 이미지	매칭 비율
1			***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 1708 # 모든매칭점 : 287 # 정상매칭점 : 12 # 정상치비율 : 0.04%
2			***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 1116 # 모든매칭점 : 272 # 정상매칭점 : 14 # 정상치비율 : 0.05%
3			***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 429 # 모든매칭점 : 187 # 정상매칭점 : 12 # 정상치비율 : 0.06%
4			***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 1716 # 모든매칭점 : 343 # 정상매칭점 : 16 # 정상치비율 : 0.05%
5			***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 2001 # 모든매칭점 : 400 # 정상매칭점 : 22 # 정상치비율 : 0.06%
6			***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 2428 # 모든매칭점 : 393 # 정상매칭점 : 19 # 정상치비율 : 0.05%
7			***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 2422 # 모든매칭점 : 392 # 정상매칭점 : 18 # 정상치비율 : 0.05%

구분	비교 이미지	매칭 이미지	매칭 비율
8			<pre> ***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 2608 # 모든매칭점 : 393 # 정상매칭점 : 14 # 정상치비율 : 0.04% </pre>
9			<pre> ***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 1701 # 모든매칭점 : 343 # 정상매칭점 : 17 # 정상치비율 : 0.05% </pre>
10			<pre> ***** # 원본비문이미지 특징점 : 1164 # 비교비문이미지 특징점 : 2001 # 모든매칭점 : 400 # 정상매칭점 : 22 # 정상치비율 : 0.06% </pre>

4.6 비문 데이터 매칭 결과

신규로 촬영된 입력 비문 데이터(Input_a)와 200개의 저장된 비문 데이터 셋(DB_Store)과 매칭을 통한 인식을 진행하였으며 특징점을 상호 추출하여 비교하였다. 저장된 비문 데이터와 동일 종의 비문을 신규로 촬영하여 비교하였으며 촬영된 Input_a의 특징점을 추출하여 저장된 DB_Store의 비문과 비교하였다. 동일한 비문을 가진 반려동물이지만 비문 촬영 시 움직임과 비문 촬영의 각도, 빛의 강도에 따라 차이가 발생하였고 이를 최소화하기 위하여 특정 영역의 동일한 형태의 객체를 추출하는 데이터 정제 과정에서 무결성을 보장하기 위한 과정을 수행하였다.



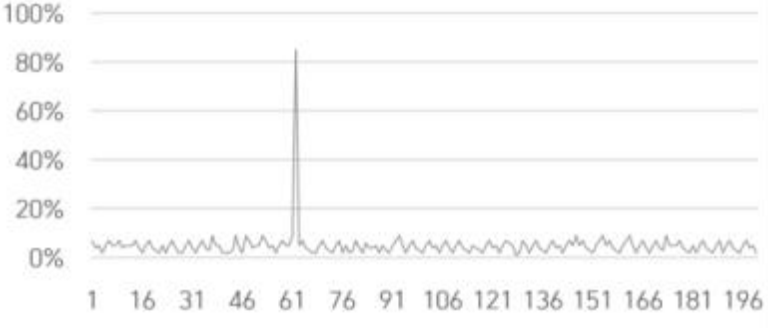
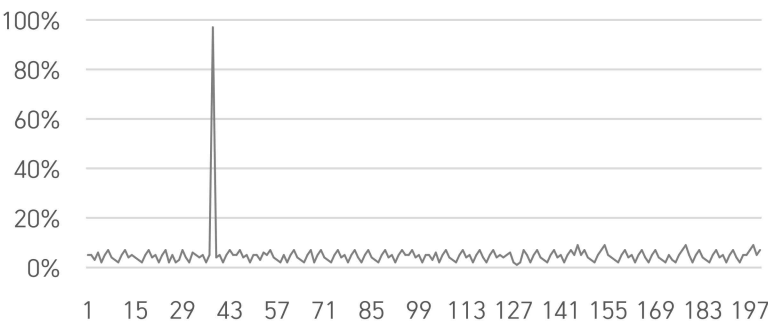
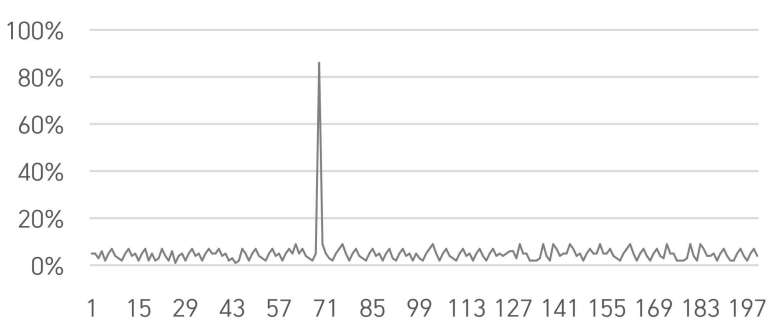
[그림 4-18] 비문 데이터 간 매칭 흐름도

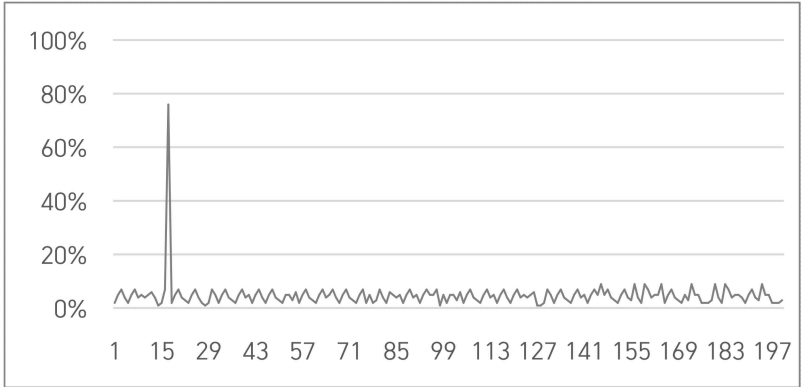
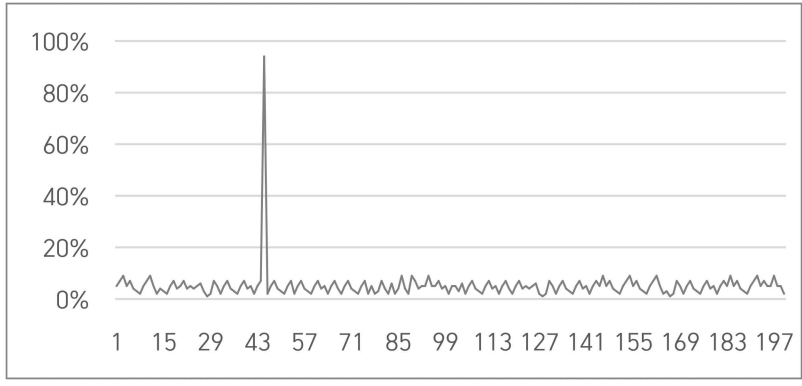
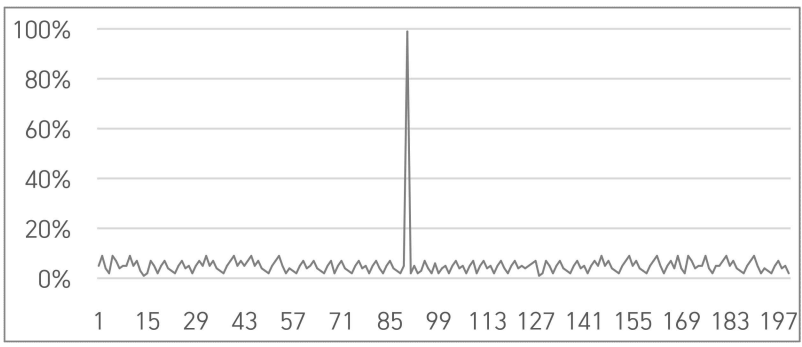
입력 비문과 저장된 200개의 비문 데이터 매칭은 pencv-contrib-python 4.5.5.64 버전과 python 3.6 기반에서 Synology DS220+ NAS 시스템에 통해 구성하였으며 비문 분석 스트리밍 서버와 데이터 관리 시스템으로 구성하여 비문 검색 API를 통해 진행하였다.

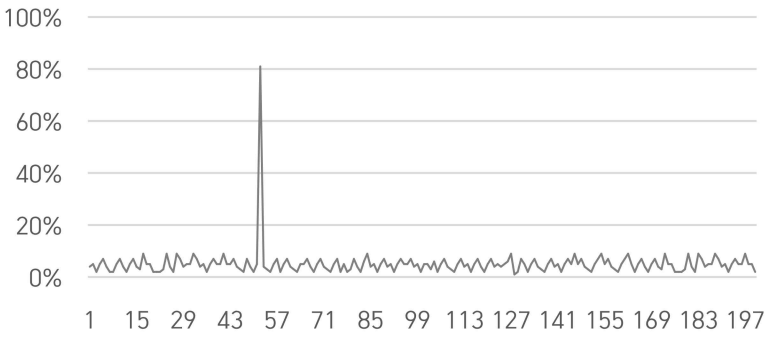
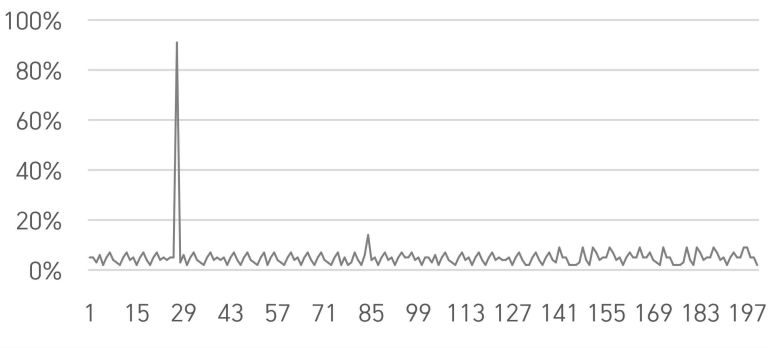
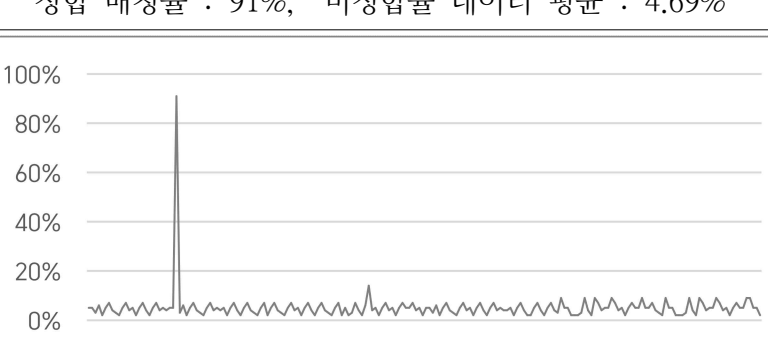
측정 결과 두 개의 비문 데이터는 [표 4-8]의 동일 비문 이미지 매칭

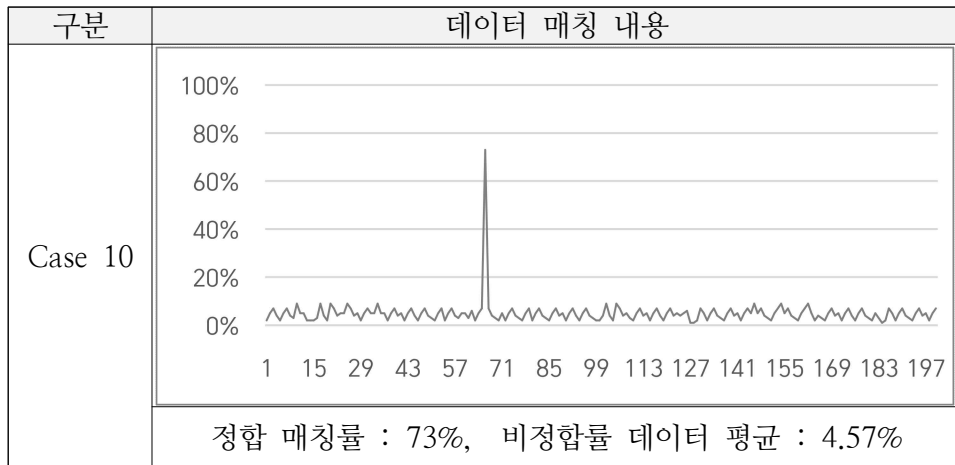
비율과 같은 매칭률을 나타내었으며 일치하지 않는 비문 데이터는 평균 4.53 ~ 4.82%로 10% 미만 대의 매칭률을 나타내었다.

[표 4-11] 데이터 매칭 그래프

구분	데이터 매칭 내용
Case 1	 정합 매칭률 : 85%, 비정합률 데이터 평균 : 4.62%
Case 2	 정합 매칭률 : 97%, 비정합률 데이터 평균 : 4.53%
Case 3	

구분	데이터 매칭 내용
	정합 매칭률 : 86%, 비정합률 데이터 평균 : 4.65%
Case 4	
	정합 매칭률 : 76%, 비정합률 데이터 평균 : 4.89%
Case 5	
	정합 매칭률 : 94%, 비정합률 데이터 평균 : 4.68%
Case 6	
	정합 매칭률 : 99%, 비정합률 데이터 평균 : 4.82%

구분	데이터 매칭 내용
Case 7	 정합 매칭률 : 81%, 비정합률 데이터 평균 : 4.74%
Case 8	 정합 매칭률 : 91%, 비정합률 데이터 평균 : 4.69%
Case 9	 정합 매칭률 : 80%, 비정합률 데이터 평균 : 4.77%



실험 결과 1개의 Input_a 데이터와 저장된 200개의 데이터셋 중 일정한 매칭 데이터의 일치율을 실험하였으며 [표 4-8]의 동일 비문 이미지 매칭 비율과 같이 동일한 데이터의 일치율은 평균 86.2%를 나타낸 반면 나머지 199개의 데이터와는 평균 4.65%로 나타났다.

기존 유사한 연구들에서 매칭과 비매칭을 구분하기 위한 특정한 수준의 threshold 값을 제안한 연구는 아쉽게도 거의 없는 실정이다. 따라서, 본 연구에서도 특정한 값을 기준값으로 제안할 수는 없는 한계는 존재한다. 그럼에도 간접적으로나마 실증 데이터셋에 본 연구의 방법론을 이용하여 매칭률을 도출해 보면, 동일한 두 이미지의 경우에는 80% 이상의 일치율이 도출되고, 그 외 다른 이미지와의 비교 시에는 일관되게 10% 미만으로 4.53 ~ 4.82 내외의 값이 나옴을 제시된 그래프 등을 보면 알 수가 있다. 이를 근거로 10% 내외를 Threshold 값으로 선정하여 실제 비문 매칭에 활용하는 것은 적절할 기준값이 될 것으로 판단된다.

V. 결론

5.1 연구의 결론

우리나라의 ‘반려동물’의 범주는 동물보호법(제32조 제1호)에 따르면 가정에서 반려의 목적으로 사육하는 개, 고양이, 토끼, 페럿, 기니피그, 햄스터를 지칭하며 국내 등록된 반려동물 수는 2015년 88만 8,000마리에서 2017년 117만 6,000마리, 2020년 154만 9,000으로 30% 이상 매년 증가 추세이며 반려동물과 관련한 시장 또한 2017년 1조 9,000억원 규모에서 2019년에는 2조 3,300억원으로 확대되었으며 2027년에는 약 6조원 대의 규모를 보이며 성장세가 가속화될 것으로 예상된다. 또한 반려동물과 함께 사는 반려 인구가 1,000만 명에 돌입하여 국내 4인 가구 중 1가구가 반려동물을 기르고 있을 정도로 발전하는 시장이다.

그러나 시장의 성장만큼 반려견에 대한 관리가 부족하며 특히 등록되어 관리하도록 법적으로 되어있음에도 실제 농림축산식품부 자료에 따르면, 2021년 기준으로 반려동물 등록률은 37.4%로 전체 반려견 등록률의 절반에도 미치지 못하고 있어 동물 등록 및 관리에 어려움을 겪고 있다. 이러한 낮은 등록률은 유기동물 문제와 연관이 있으며 2021년 한해에만 12만 마리의 반려동물이 버려진 것으로 추정되고 있어 심각한 사회 문제로 대두되고 있다. 따라서 체계적인 반려동물의 관리가 필요하며 본 연구에서는 생체 인식 기반 비문을 활용한 인식률 고도화 방안을 목표로 연구하였다.

비문 이미지는 사람의 지문과 같이 독특한 구조의 형태로 비문의 유일성을 기반으로 ORB 알고리즘 기반에 BEBLID Descriptor를 활용하여 비문의 전역적 특징을 추출한 후 저장된 DB의 데이터와 비교하여 일치성의 매칭률에 대한 평가를 새롭게 하였다. 이를 위해 비문 정보를 ROI 영역을

설정하여 특징점을 추출하여 이진화된 모델로 변경하고 결과물로 생성된 변환 행렬을 적용하여 상호 비교하며 BEBLID의 기술자(Descriptor)의 최적값을 산정하였다.

본 연구에서는 ORB 알고리즘은 객체 인지 방식의 대표적 문제점인 영상의 크기가 달라지는 경우 특징점으로 검출되지 않을 수 있는 단점을 개선하였으며 정확도를 높이는 방안으로 BEBLID Descriptor 값을 적용하였다. ORB 알고리즘을 개선한 BEBLID 접근법에 기반한 비문 패턴을 이용한 새로운 식별 방법을 비문 등록의 한 방법으로 제안하며 인식 및 매칭률 향상을 위해 알고리즘 매칭의 주요 요소인 기술자의 개선을 중점적으로 논하였다. ORB 알고리즘의 코너 검출 방법을 통해 특징점을 추출한 뒤, 특징점의 방향성분을 계산하고 데이터 분석을 하였다. ORB 알고리즘을 이용하여 특징점의 방향 정보를 계산하였으며 객체의 크기 조정, CLAHE 및 중복 일치 제거 등을 통해 기초 데이터를 구성하였다. 객체의 크기 조정 및 CLAHE는 이미지의 일치 성능을 높이며 중복성 제거는 서로 다른 비문 이미지의 매칭률을 향상시켰다.

본 연구에서는 코너 검출 방법을 통해 특징점을 추출한 뒤, 특징점의 방향성분을 계산하고 데이터 분석 기술을 적용하였으며 이진 기술자를 사용하여 특징점의 방향 정보를 정확한 값으로 계산 후 추출 키포인트 변환과 BEBLID의 Descriptor 값을 0.75~2을 적용하여 실험하였다. 그 결과 BEBLID Descriptor 값 1을 사용하여 Source 데이터와 DB에 저장된 데이터의 추출된 특징점을 비교하면 76.24%의 매칭률을 얻었으며 이는 ORB 알고리즘의 매칭은 66.82%보다 9.42% 나은 결과이다. BEBLID의 Descriptor K값을 가변적으로 실험 결과, 적정값은 0.75로 도출되었다. 이 경우 특징점 인식률은 87%로 기존 76.24%에서 10.76% 개선되어 전체적으로 ORB에 비해 20.18% 향상됨을 알 수 있었다. ORB 알고리즘 기반 비문인지 서비스를 위해 비문 데이터를 생성하였으며, 데이터 허브에 등록된 비문 데이터셋 정보를 프로토타입 서비스 실증을 통해, 매칭률을 확인

하였으며 100%의 정확도를 나타내었다. 다만 데이터의 양이 200여 건으로 소량이며 매칭률이 10% 이상인 경우 경험치에 의해 동일 비문으로 인지하였으며 이에 대한 적정성에 대한 추가적인 연구가 필요하다.

본 연구는 비문 관련 데이터의 수집, 저장 및 분석을 통한 비문 인식 기술의 매칭률 정확도 향상에 관한 것이다. 기존의 매칭 알고리즘인 ORB에 BEBLID 기술자를 적용하였으며, 특히 BEBLID 기술자는 데이터 세트의 인식률 향상 및 매칭률 정확도에 영향을 미치는 중요 요소이다. 다양한 비문 데이터는 디지털 전환(Digital Transformation)을 기반으로 다양한 분석 및 의사결정을 지원하며 이를 통해, 더 나은 비즈니스 결과 추구가 가능하게 된다.

본 연구에서는 객체인지의 다양한 알고리즘을 비교 분석하였으며 인식률 제고 알고리즘인 BEBLID를 반려견의 비문인지에 접목함으로써 매칭률 기반 고도화 연구를 진행하였다, 비문 인식률 향상 기술 구현을 통해 반려견의 체계적인 관리와 유기로 인한 사회 문제 해소가 가능하며 반려동물의 비문 패턴 이미지의 정확도 향상은 동물등록제의 등록률을 향상시킬 것이다.

또한 ORB 알고리즘 기반 BEBLID Descriptor를 활용하여 비문 인식 성능을 향상시킬 수 있는 방법을 이용하면 기존 동물 등록 방식인 IC 칩 삽입 및 목걸이 태그 방식을 대체하는 새로운 등록 방식으로 적용이 가능할 것으로 예상된다. 반려동물 관련된 산업 규모의 확대와 서비스는 점차 확대되고 전문화 될 것으로 전망되며 비문 기반 보험시장, 관리 솔루션 등 다양한 사업과의 연계가 가능할 것으로 예상된다. 관련된 비문 데이터를 활용하여 다양한 적용이 가능하며 특히 AI·빅데이터 산업에서의 양질의 비문 데이터는 인식 과정을 거쳐 등록률 제고 및 각종 연구 성과로 연결할 수 있을 것이다. 현재의 시대에서 데이터는 중요한 디지털 자산으로 정확도를 기반으로 생성된 비문 데이터는 등록률 제고 및 반려견과 함께하는 우리 삶의 변화를 줄 것이며 관련한 동물 관리 시스템 구현 및 솔루션 기반 연계 사업으로 발전될 것으로 기대된다.

5.2 연구의 한계점 및 향후 연구방향

5.2.1 연구의 한계

본 논문은 기존 매칭 알고리즘인 ORB에 추가로 BEBLID Descriptor를 접목하여 동물의 비문 이미지에 적용하였으며 이를 통해 비문 인식률이 기존 방식에 비해 우수함을 입증하였다. 향후 반려동물의 시장성을 고려할 때 다양한 분야에 적용이 가능할 것이다. 이러한 결론에도 불구하고 본 연구에는 몇 가지 한계점을 가지고 있으며 구체적으로 다음과 같다.

첫째, 본 연구에서 사용한 데이터의 양은 200건으로 적지 않지만 보다 더 많은 양의 데이터를 확보한 환경에서도 마찬가지로 매칭 시간 및 정확도가 유지되는지에 대한 추가 연구의 필요성이 있다. 즉, 더 많은 비문을 대상으로 알고리즘을 접목하여 인식률을 평가하고, 성능 향상이 가능한지에 대한 연구가 필요하다.

둘째, 본 연구에서는 비교되는 두 개의 이미지에 대하여 inliers1 및 inliers2 변수 간에 매칭되는 일치점이 10% 이상일 경우 동일한 이미지인 것으로 판단하도록 Threshold 값을 설정하였다. 이는 객체 인지 관련 방법론에서 인식에 대한 기준치가 필요하며 기준치 아래는 비매칭된 데이터로 반대로 기준치 이상은 매칭된 데이터로 분류한다. 본 연구의 실험 결과 비매칭 데이터의 평균값이 4.75%로 나타나 10%를 Threshold 값으로 산정함에 문제는 없는 것으로 사료되나 추가 반려동물과 추출된 비문 데이터 분석 결과를 기반으로 보다 세밀한 Threshold 값에 대한 연구가 필요할 것으로 판단된다.

마지막으로 본 연구는 개의 비문을 대상으로 실험하였으나 고양이와 같이 작은 비문을 가진 경우에도 인식률을 실험해 볼 필요가 있다. 반려동물은 크기가 작고 움직임이 활발하여 비문 이미지를 획득하기가 어려운 점이 많아 좀 더 작은 객체의 비문에 적용하는 추가 연구가 필요하다.

5.2.2 향후 연구 방향

객체 인식은 정보를 인식하는 시각 정보를 인공지능 기반 컴퓨터가 대신하여 분석하고 해석할 수 있도록 하는 연구 분야이다. 이는 얼굴 인식, 영상인식, 자율 주행, 제조업, 보안 등에 다양하게 활용되고 있다. 이러한 중요성을 인지하여 객체 인식 분야는 다양한 분야로 확대 적용되고 있는 추세이며 대표적으로 비문 기반 생체인식 부문을 들 수 있다.

완벽한 비문 기반 생체인식을 위해서는 ‘유일성, 영속성, 획득성’과 같은 3가지 기술적 요소를 고려하여야 한다. 유일성은 동물의 비문 객체 정보가 다른 객체의 생체정보와 겹치지 않아야 하며 영속성이란 비문 생체정보를 매번 등록하지 않아도 계속해서 사용할 수 있음을 의미한다. 그 과정에서 별다른 불편함을 느끼지 않은 채 다양한 환경에서 손쉽게 사용할 수 있어야 하는 획득성이 있어야 한다.

현재 비문 인식은 유일성, 영속성 그리고 획득성의 3가지 필수요소 중 유일성 및 영속성은 과거 연구에서 증명이 되었으며 획득성은 얼마나 많은 반려동물의 비문을 저장하고 인식에 성공적이었는가가 중요한 요소이다. 과거 비문 개발 사례는 소와 같은 대형 동물 위주였으며 개와 같이 작은 비문은 연구 경우가 많지 않았다. 본 연구에서는 200개의 데이터 셋으로 데이터베이스를 구축하였으나 향후 더 많은 비문 객체를 확보하여 실험하여야 할 것이다.

또한 비문 데이터의 정확한 인식을 위한 평가지표가 명확히 산출되어야 한다. 비문 대상 생체인식 기술의 대표적인 평가지표는 등가 오류율(Equal Error Rate, EER)과 등록 실패율(Failure To Enroll rate, FTE), 인증 실패율(Failure To Acquire rate, FTA) 등이 있다.

향후 평가지표를 기반으로 반려동물 생체인식 기술 표준화가 필요하며 공인 시험인증센터 등 전문기관에서 생체인식 공인 DB 구축 가이드라인, 생체인식 시험평가 체계에 대한 기준을 마련하여 지속적인 실험이

가능하도록 관련된 평가지표가 마련되어야 한다. 국내 비문 관련 연구에서 각자의 데이터베이스를 근거로 관련 지표가 입증 가능한 수준이라 밝히고 있지만 이를 객관적으로 뒷받침해 줄 표준이 없기 때문에 관련된 표준화가 시급하다. 대표적으로 매칭의 정확도를 나타내는 임계치 값인 Threshold 10%의 적정성을 들 수 있다.

현재 국내 동물등록제 주무 부처인 농림축산식품부는 비문 인식 기술의 동물등록제 입법과정을 규제 샌드박스를 통해 제공 예정 중이며 이를 위해 비문 관련 표준화와 기술 변별력을 밝혀내는 기준 수립을 준비 중이다. 향후 비문 관련된 산업 규모의 확대와 서비스는 점차 전문화될 것으로 전망되며 반려동물 관리 시스템 구현 및 솔루션과 연계하는 사업은 확대 발전될 것으로 기대된다.

반려동물 등록관리 플랫폼은 실시간성 데이터를 수집·저장하며, 데이터 스트림으로부터 빅데이터 분석, 통계 분석, 인공지능(AI) 모델 등을 통해 데이터의 실시간 정제 또는 가치향상이 가능하다, 이 과정에서 정제된 비문 데이터 또는 모델 실행 결과 등이 데이터 기반으로 분석 가능해짐으로써, 관련 데이터를 적극 활용하는 반려견 관리체계 마련이 가능해질 것으로 사료된다. 비문 이미지에서 추출된 특징 설명자를 사용하여 매칭 수행하며 정보 공유를 위한 원장 관리 기능과 저장된 원장에 정보를 기록 및 조회를 위한 스마트 관리 기능을 구현하며 프라이버시로 기밀성 보장 기반 데이터 거래 기능 구현 등이 가능하게 될 것이다.

본 연구에서는 기존의 특징점 추출 알고리즘인 ORB 알고리즘 기반에 BEBLID 모듈을 반영하여 인식률을 고도화하였으며 비문 이미지의 추출된 특징 설명자를 이용하여 반려견 정보 기록, 조회 등이 가능한 플랫폼으로 발전될 것이다. 아울러 동물 등록률 향상 효과와 더불어 Pet 서비스의 발전 모델을 정립하고 실질적 서비스 제공이 가능하도록 사업적인 대안을 제공할 것으로 기대된다.

참 고 문 헌

1. 국내 문헌

- 김용현, 박원표, 노명철, 신종주. (2020). Learning Latent Groups and Constructing Group-based Representations for Face Recognition, 10-14.
- 김기연, 권호영. (2020). 2019년 반려동물 보호와 복지에 관한 실태 조사. 농림축산검역본부
- 김종복, 장동화, 양가영, 권경석, 하태환, 이준환. (2022). 딥러닝을 활용한 반려견 비문 이미지 품질 분류 연구. 한국산학기술학회 논문지, 23(4), 34-40.
- 최형인, 이윤석, 신현진, 이성진. (2021). The Formation and Invariance of Canine Nose Pattern of Beagle Dogs from Early Puppy to Young Adult Periods, 11(9), 120-128.
- 장도화, 권경석, 김중곤, 양가영, 김종복. (2020). 비문 기반의 개식별방법 연구, 농촌진흥청 국립축산과학원
- 최현인, 이윤석, 신현준, 최수진, 한창용, 권송화. (2021). The Formation and Invariance of Canine Nose Pattern of Beagle Dogs from Early Puppy to Young Adult Periods Animals, Vol 11(9).
- 정제창, 이민정. (2017). 개체 특징점 추출 방법 및 장치, 이를 이용하는 개체 인식시스템
- 최형인, 이성진, 문환표, 위남식, 김대훈, 권송화 (2021). Seeded using Model and Distributed Biometric Template Storage and Matching, 849-851.
- 한창용, 권송화, 최현인 (2018). Seeded using model and the statistical nature of human iris templates, 98-102.

황세웅(2023). 데이터 분석가가 반드시 알아야할 모든 것.
 황원경 & 이신애. (2023). 2023 한국반려동물 보고서.
 탁기현, 방효충. (2015). ORB를 이용한 실시간 목표물 인식 및 추적. 한국항공우주학회, 1068-1071.

2. 국외 문헌

Alahi, A., Ortiz, R., & Vanderghenst, P. (2012). Freak: Fast retina keypoint. Computer Vision and Pattern Recognition Conference, IEEE. 510-517

Andrew, W., Greatwood, C., & Burghardt, T. (2017). Visual localisation and individual identification of holstein friesian cattle via deep learning, IEEE International Conference on Computer Vision Workshops (ICCVW), 2850-2859.

Awad, A., Zawbaa, H., Mahmoud, H., Nabi, E., Fayed, R., & Hassanien, A. (2013). A robust cattle identification scheme using muzzle print images. Federated Conference on Computer Science and Information Systems, 529-534.

Baranov, A., Graml, R., Pirchner, F., & Schmid, D., (1993). Differences and intra-breed genetic variability of dermatoglyphic pattern of cattle, Animal Breed Genet, 385~392.

Barry, B., Gonzales, U., McDonnell, K., Butler, F., & Ward, S., (2007). Using Muzzle pattern Recognition as a Biometric Approach for Cattle Identification, Trans. ASABE, 1073-1080.

- Bay, H., Tinne, T., & Luc, V., (2008). Speeded-up robust features. *Computer vision and image understanding*, vol.110(3), 346-359,
- Bengio, Y., Courville, A., & Vincent, P., (2013). Representation Learning A Review and New Perspectives, IEEE Trans : PAMI Press
- Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Binary robust independent elementary features, *Computer Vision-ECCV*, 778-792.
- Coldea, N. (1994). Nose prints as a method of identification in dogs, *Vet. Q*(1), 60-60.
- Evans, H., & Lahunta, A. (2019). *Miller's Anatomy of the Dog*, Vol 4, 145-149.
- Fischler, M., & Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM*, 381-395.
- Horning, A., Mckee, H., Keller, E., & Smith, K. (1926). Nose Printing Your Cat and Dog Patients, 432-435.
- Gaber, T., Tharwat, A., Hassanien, A., Snasel, V., (2016). Biometric cattle identification approach based on Weber's local descriptor and AdaBoost classifier, *Comput. Electron. Agric.* Vol. 122, 55-66.
- Ismail, A., & Hassaballah, M., (2021). Bag-of-Visual -Words for Cattle Identification from Muzzle Print Images
- Kumar., & Santosh., (2018). Deep learning framework for recognition of cattle using muzzle point image pattern,

Vol 116, 1-17.

- Lago, S., Ghesn, S., José, M., & Baumela, L., (2020). BEBLID: Boosted efficient binary local image descriptor, *Pattern Recognition Letters* Vol 169, 366-372.
- Lowe, D. (2004). Distinctive image features from scale- invariant keypoints, *International journal of computer vision*, vol.60, 91-110.
- Minagawa, H., Fujimura, T., Ichiyanagi, M., Tanaka, K., & Fangquan, M. (2002). Identification of beef cattle by analyzing images of their Muzzle patterns lifted on paper, *Publ. Society Agric Information*, 596-600.
- Mei, W., & Weihong, D. (2018). Deep Face Recognition, *computer science*
- Mustaqee, (2022). Advanced Study of Emotion Recognition Based on Speech Signals using Deep Learning, 19-37.
- Ning, Y., Wu, Z., Xing, C., & Zhang, L. (2019). A review of deep learning based speech synthesis, *Appl. Science*, 40-50.
- Noviyanto, A., & Arymurthy, A., (2013). Beef cattle identification based on Muzzle pattern using a matching refinement technique in the SIFT method, *Comput. Electron. Agric.* Vol 99, 77-84.
- Petersen, W. (1922). The Identification of the Bovine by Means of Nose-Prints, *Dairy Science*, 249-258.
- Pisano, E., Zong, S., Hemminger, B., DeLuca, M., & Muller, R. (1998). Contrast Limited Adaptive Histogram Equalization image processing to improve the detection of simulated spiculations in dense mammograms, 193-200.

- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G., (2011). ORB: an efficient alternative to SIFT or SURF, Computer Vision, IEEE International Conference, 2564-2571.
- Shuaishuai, P., Qicheng, Y., & Tong, W., (2022). Proceedings of SPIE, An improved GMS image feature matching algorithm based on BEBLID descriptor. The International Society for Optical Engineering.
- Tareen, S., & Saleem, Z. (2018). comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. Mathematics and Engineering Technologies, 1-10.
- Tharwat, A., Gaber, T., Hassanien, A., Hassanien, H., & Tolba, M. (2020). Cattle Identification Using Muzzle Print Images Based on Texture Features Approach, *Materials Science and Engineering*, Vol. 853.
- Tong, H., Li, M., Zhang, H., & Zhang, C. (2004). Detection for digital images using wavelet transform. IEEE International Conference on Multimedia and Expo Vol. 1, 17-20.
- Wang, N., Chen, Y., Zhang, X., XuhongChiong, Z., & Raymond, M. (2020), A-BEBLID: A Hybrid Image Registration Method for Lithium-Ion Battery Cover Screen Printing, Institute of Electrical and Electronics Engineers.
- Zibo X. & Dongyu P.(2023). PTZ Active Vision Method Based On BEBLID, Journal of physical, 6-7
- Zhangyong, L., Shen, S., Ge, C., & Li, X. (2018). Cow individual identification based on convolutional neural network. Computing and Artificial Intelligence, 21-23.

- Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object detection in 20 years, A survey Proceedings of the IEEE.
- Zuiderveld, K. (1994). Contrast Limited Adaptive Histogram Equalization, : Academic Press Professional Inc., Graphics Gems IV, 474-485.

ABSTRACT

Research on Derivation of Optimal BEBLID Descriptor Value for Muzzle Pattern Utilization and Advancement of Object Recognition Matching Technology

Yang, Jeon-Seong

Major in Smart Convergence Consulting

Dept. of Smart Convergence Consulting

The Graduate School

Hansung University

In South Korea the number of households with pets has grown continuously, reaching 3.13 million according to Statistics Korea's 2020 Population and Housing Census. The rise in this number is also related to the increase in single-person households (34.5% of the South Korean population, 2022 Population and Housing Census, Statistics Korea). These trends, along with the spreading recognition of pets as a member of the family, point to the need for systematic pet

management. In response, the South Korean government has introduced the animal registration system under the Animal Protection Act. Currently, two methods are available for pet registration in South Korea: having pet identification chips implanted in their pets or attaching pet registration tags to their pets' collars. Neither of the methods have been widely implemented, with pet owners feeling hesitant about the implantation method due to ethical issues and the collar tag method due to its inconvenience, such as loss and damage.

In light of such low response among pet owners, biometric technology has been gaining attention as a more convenient and less intrusive method for registering pets. Specifically, biometric technology identifies faces or parts of the body from standardized images based on their distinct features, which are called keypoints such as tilt vector and color difference. Existing techniques for recognizing objects in specific areas of an image include Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Binary Robust Invariant Scaling Keypoints (BRISK), and Oriented FAST and Rotated BRIEF (ORB). In existing algorithms, low image resolution and the variations in the images' scale and the rotation angle tend to lower the object recognition rate.

The experimental procedure consisted of a total of 10 steps, including constructing a muzzle pattern database (DB), extracting features, post-processing the dataset, matching the dataset, and analyzing results. The recognition and matching experiment with 200 data samples resulted in an over 70% matching rate, proving the feasibility of this method to substitute existing methods.

A minimum 85% matching rate was set as the standard for

determining the optimal value of the BEBLID descriptor. The object recognition rate of small images like muzzle patterns is low, especially when the image includes many fine details, such as wrinkles. Since accurate image recognition and analysis is important for object recognition technology, the optimal value of BEBLID descriptor is derived to resolve ORB algorithm's problem and utilize its excellent feature extraction ability.

This study's findings on improving the object recognition rate are expected to contribute to the implementation of the available methods for registering pets

This study proposed a new way to recognize muzzle patterns by applying BEBLID to the ORB algorithm and also derived BEBLID's optimal K value, which is the scale coefficient and the main variable of BEBLID, for improving the muzzle pattern recognition and matching rate. Specifically, the ORB algorithm was used to extract the keypoints of the datasets using the corner detection method and calculate the directional information of the keypoints. Then, BEBLID was applied using variant K values from 0.25 to 2 to investigate the improvements in matching accuracy.

A matching rate of 76.24% was obtained by comparing the extracted keypoints of the source and stored data using the ORB algorithm applied with BEBLID (K value=1), which is 9.42% higher than the matching rate of ORB alone (66.82%). Also, BEBLID's optimal K value was determined to be 0.75, achieving a matching rate of 87%, which is 10.76% higher than when the K value is set to 1 and 20.18% higher than when ORB is applied alone. The improved muzzle pattern recognition performance reported in this study is

expected to contribute to introducing muzzle patterns as an alternative pet registration method to implanting IC chips and attaching registration tags to collars.

Despite its contributions, this study also contains some limitations. First, this study used 200 muzzle patterns as its data, which is not a small number, but a larger data may be required to draw generalized conclusions on the time it takes to process the matching and matching accuracy. Second, this study set the threshold value of 10% for judging two images as identical. That is, if 10% or more of the keypoints were matched between inlier 1 and inlier 2, the two images were considered the same. Although this threshold value is in line with most methodologies for object recognition techniques, it may be possible that a different threshold value may be more optimal for muzzle pattern recognition. It may be worthwhile to explore the determination of the optimal threshold value for muzzle pattern recognition in future research. Finally, this study experimented with muzzle patterns of dogs, but further studies are required to test the recognition and matching rate on muzzle patterns that are smaller, like those of cats.

The researchers hope that further developments will be made in the models for muzzle pattern recognition to enable positive changes in the systematic caring and managing of the pet population.

【Key words】 BEBLID algorithm, Descriptor, Biometrics, Muzzle pattern, Keypoint Detection