ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20231120001128
Original scientific paper

Optimizing Muzzle Pattern Identification through Improved Pattern Recognition Techniques

Jeon Seong YANG, Seok Kee LEE*

Abstract: The global pet industry and market have been thriving in recent years, and the rise in pet population has led many countries to implement policies and systems for their management. The South Korean government has also introduced an animal registration system under the Animal Protection Act, but only about 37% of the country's pet population are registered in the system due to the inconvenient methods of registration. Biometric technology, in general, is used to identify individuals based on accurate object recognition. This paper applies Boosted Efficient Binary Local Image Descriptor (BEBLID) to the commonly-used ORB algorithm with the goal to improve the recognition and matching of muzzle pattern data and to derive the optimal value of the BEBLID scale coefficient (*K*) for muzzle pattern recognition. A total of 200 muzzle patterns were collected from dogs to use as the data for analysis. To demonstrate the superiority of the proposed method, the ORB algorithm was used as a benchmark. The matching rate achieved when BEBLID's *K* value was set to the default value of 1 was 76.24%, compared to the 66.82% matching rate of the ORB algorithm. Also, the optimal *K* was determined to be 0.75, achieving an 87% matching rate, after testing with variant *K* values from 0.25 to 2. Overall, this study demonstrates that by applying BEBLID to traditional ORB algorithms using the optimal scale coefficient value, the recognition and matching rate can be significantly improved. The commercialization and practical application of the muzzle pattern recognition technique proposed in this study is expected to contribute to improving the pet registration rate, which has been stagnant despite its necessity, and the management of pet populations.

Keywords: BEBLID algorithm; biometrics descriptor; keypoint detection; muzzle pattern

1 INTRODUCTION

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 has 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 [1]. Specifically, biometric technology identifies faces or parts of the body from standardized images based on their distinct features, which are called key points such as tilt vector and color difference [2]. 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 [3]. 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.

2 LITERATURE REVIEW

2.1 Principles of Object Recognition Technology

The techniques for identifying an object in an image can be divided into object detection and object recognition. Object detection captures the presence of the object in an image as well as its location. Thus, object detection can identify multiple objects in an image and images that are the same [4]. Object recognition, on the other hand, identifies the objects in an image, representatively using deep learning and machine learning algorithms. That is, object recognition technique identifies objects based on an understanding of the image that has been acquired naturally through accumulated learning [5]. Object recognition models have been developed as deep learning or weight update models, which derive predicted values from a hierarchy of layers consisting of an input layer, hidden layers, and an output layer. The loss of input is calculated using a loss function, which defines the difference between the predicted and actual outputs, and improvements are made in the data recognition rate by gradually minimizing the loss and modifying the weighted values of the model. The data is repeatedly learned for optimal recognition of objects. This study focuses on object recognition, specifically the application of BEBLID to the ORB algorithm.

2.2 Overview of Existing Algorithms

2.2.1 ORB Algorithm (Oriented FAST and Rotated BRIEF)

The ORB algorithm was developed by OpenCV Labs to replace the SIFT and SURF algorithms. Existing keypoints detection methods have difficulties in extracting keypoints depending on the size of the object, and their recognition rate decreases when analyzing images of varied scales and directions. Thus, there was a need for an algorithm that reflects the information on the directions of keypoints for accurate extraction. ORB extracts keypoints using the FAST corner detection method, calculates the directional component of the keypoints, then computes the binary descriptors using the BRIEF algorithm. The principle behind ORB algorithm's keypoint extraction is to randomly select the images to compare using Gaussian distribution, then performing a pair-wise comparison of brightness to put together descriptors. Then, a binary test renders the information on the direction of the keypoints composed of n-dimensional binary strings of n bits, noted as $des_{n,\ BRIEF}$. $des_{n,\ BRIEF}$ with n bits through binary test. As such, ORB boasts a high accuracy and is able to detect keypoints even when the size of the image changes [6].

$$y\left(p;x,y\right)=\begin{Bmatrix}1:I\left(x\right)<I\left(y\right)\\0:I\left(x\right)\geq I\left(y\right)\end{Bmatrix} \tag{1}$$

$$des_{n,BRIEF}\left(p;i\right)=2^{i-1}y\left(p;x_i,y_i\right) \tag{2}$$

where $1\leq I\leq n$, 'p': image patch corresponding to a window; $I(x)$, $I(y)$: the brightness of the pixels at points x and y in image patch p.

To extract keypoints from images of multiple scales, ORB applies a multiscale image pyramid. This makes it useful for analyzing muzzle patterns, as the elements surrounding the keypoints of muzzle patterns are perceived in the form of individual pyramids. The keypoints for different scales are extracted in advance from an image that is smaller than the original, thereby making them scale-invariant. The ORB algorithm is a combination of the FAST and BRIEF algorithms and is often used due to its very fast speed and good performance compared to SIFT and SURF. SIFT has the advantage of being scale-invariant, its extraction process is complicated and requires a certain amount of time to compute the relative direction [7]. BRIEF algorithm shows improved recognition speed by minimizing descriptor generation and comparative computations and makes efficient use of memory as it uses binary bits [8].

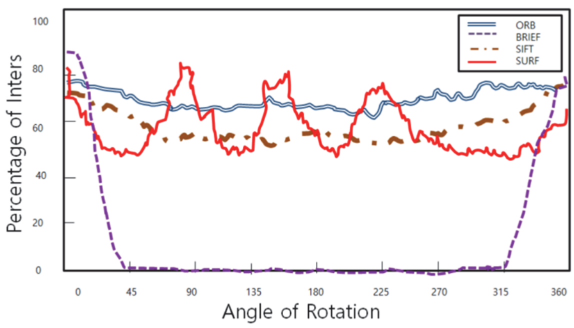

Figure 1 Matching performance of recognition algorithms

Fig. 1 shows the matching performance of four algorithms (ORB, BRIEF, SIFT, and SURF) on a rotated image with Gaussian noise. The results show that SIFT is more reliable than SURF, with ORB performing best with a matching performance of over 70% [9].

2.2.2 BEBLID (Boosted Efficient Binary Local Image Descriptor)

BEBLID was introduced to overcome the limitations of existing algorithms (e.g., SIFT, SURF), such as high computational costs and the need for large amounts of memory [10]. The core idea of BEBLID is to use boosted classifiers to select and use subsets of functions. BEBLID takes a patch-based approach in which images are split into small patches to extract binary feature descriptors independently from each patch for efficient use. BEBLID can be applied to the ORB algorithm to overcome the representative shortcoming of object recognition technology, which is being unable to detect keypoints when the size of the image changes. In this study, BEBLID was applied to improve the recognition rate of the existing ORB algorithm to improve image matching accuracy and reduce processing time. To achieve this, setting an appropriate scale coefficient is critical in order to use BEBLID efficiently. If BEBLID's scale coefficient (K), which is a binary image descriptor, is too large, the keypoint matching rate drops. Conversely, if K is small, more pyramid levels must be constructed, which slows down the computation speed. Hence, BEBLID's scale coefficient was also experimented to determine the optimal value for improving the muzzle pattern recognition and matching rate.

2.3 Existing Studies on Muzzle Pattern Recognition

Muzzle pattern refers to the wrinkles on the nose of animals that take different shapes depending on the animal's sweat glands [11]. Multiple studies and experiments have proven that muzzle patterns are unique codes, just like the human fingerprint. Representatively, Minagawa et al. (2002) studied the binary conversion and morphological approach for extracting keypoints characterized by pixels from muzzle patterns, but the study only achieved a keypoints matching rate of about 30% [12]. Barry et al. (2007) developed a method for identifying muzzle patterns by pre-processing keypoints and using a unique surface algorithm [13]. By dividing and matching the data into segments, they raised the accuracy to 98.5%, but the study only used a relatively small comparative dataset and only included the muzzle patterns of cows, which are relatively larger than other animals. Noviyanto and Arymurthy (2013) applied the SIFT algorithm to 160 datasets of cow muzzle patterns to derive keypoints [14]. They reduced abnormal values through post-processing and studied the robustness of the algorithm against changes in image scale and rotation. This study further used SURF to identify the muzzle patterns and reported a recognition accuracy of more than 90%. More recently, research on muzzle patterns has looked toward artificial intelligence (AI)-based deep learning. Kumar et al. (2018) analyzed a total of 400 cow muzzle patterns using a deep learning model and concluded that a standardized method was needed considering the difficulty of recognizing muzzle patterns when their sizes and the cow breed are different [15]. The major studies on muzzle patterns since the 2000s are listed in Tab. 1.

Table 1 Representative muzzle pattern research

Year	Author	Dataset	Technology used	Accuracy
2002	Minagawa et al.	Dog/43 Sets	Paper-based comparison method	30%
2007	Barry et al.	Cow/29 Sets	Eigenvalues + Segmentation	98.5%
2013	Awad et al.	Cow/15 Sets	SIFT + RANSAC	93.3%
2016	Gaver et al.	Cow/31 Sets	WLD + ABD	99%
2017	Kumar et al.	Cow/400 Sets	Deep Learning DSN Model	95.9%

3 RESEARCH METHODOLOGY
3.1 Experimental Procedure

This study applied biometric technology to the muzzle patterns of animals, specifically dogs, as a way to boost the pet registration rate. A muzzle pattern refers to the various types of wrinkles that make up the shape of an animal's nose. Like the human fingerprint, each individual animal has a unique muzzle pattern. ORB algorithm provides better performance for the representative issue faced by object recognition algorithms, that is, the inability to detect features if the size of image changes. However, its reliability for dealing with scale variance is still relatively less strong. Thus, a different extraction algorithm should be applied depending on the field of application. This study attempted to achieve faster and more accurate muzzle pattern recognition by applying a BEBLID module to the keypoint detecting ORB algorithm. For the study, a significant amount of time and effort was spent on securing the quality data, including taking photos of muzzle patterns directly and visiting relevant organizations to obtain existing muzzle pattern data. Experiments were performed by applying variable values of the BEBLID descriptor using the ORB algorithm to derive the optimal value for muzzle recognition and matching and to discuss how object recognition technology can be improved for muzzle recognition. Image processing for muzzle recognition was performed through Image Resize, CLAHE, RANSAC, and DMR processes [16]. The keypoints were extracted using SIFT, SURF, BRISK, and ORB, and the descriptor, BRIEF, and BEBLID provided by default were calculated for each keypoint. The BEBLID descriptor, which allows the user to set the scale factor, was variably applied from 0.25 to 2 for comparison. BEBLID descriptor's scale factor refers to the image pyramid, which can have various versions of the same image in different resolutions, with each level depending on the scale of the original image. The scale factor determines the relative scale between two successive pyramid levels. It is possible to adjust the performance and speed of keypoint detection by adjusting the scale factor of the image pyramid, where a lower scale factor produces more accurate results, but can be more costly to compute. Thus, it is important to derive the most suitable value [17]. BEBLID was applied to the ORB algorithm to compare the collected muzzle pattern data and the muzzle pattern data stored in the database through a process consisting of ten steps in total [18]. The ten steps were as follows: 1) Muzzle pattern data collection, 2) Muzzle pattern dataset construction, 3) Data augmentation, 4) Image resizing, 5) Contrast Limited Adaptive Histogram Equalization (CLAHE) application, 6) Feature extraction, 7) Muzzle pattern data matching, 8) Random sample consensus (RANSAC) application, 9) Duplicate matching removal, and 10) Analysis of results. These steps involve collecting and improving the image data by removing noise and clarifying their structure, then extracting keypoints and feature descriptors for pair-wise comparison, and finally removing duplicates and evaluating the obtained results. The detailed explanation for each step is shown in Fig. 2.

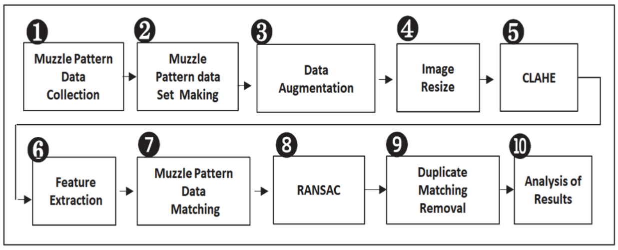

Figure 2 Muzzle pattern recognition and matching process

To start, muzzle pattern data were collected from actual pets using a smartphone camera that takes 4608000 pixel photos at a 1440 × 3200 resolution. Multiple consecutive photos were taken as the dogs moved during the photo shooting, and the most suitable images were selected as the data [19]. Also, the photos were taken indoors to avoid the reflection of direct sunlight. The muzzle patterns of 200 dogs were obtained at a distance of less than 10 cm. In the following muzzle pattern dataset construction step, these muzzle patterns were analyzed to create a dataset, that is, a matrix of data that describes each data by multiple attributes, such as entity, object, and individual [20]. The regions of interest (ROI) in the initially collected data of 216 muzzle patterns were tested by changing its resolution from a minimum of 180 × 170 pixels to a maximum of 841 × 825 pixels. As a result, 16 muzzle patterns that showed degradation at the pixel count of 549 × 515 were excluded from the dataset, thereby leaving 200 muzzle patterns for experimentation.

Table 2 Data collection and Resolution

Resolution	Pixels	Training (Source Data)	Test (Confirmed Data)	Confirm ratio
180 × 170	30600	216	216	100%
290 × 280	81200	216	204	94%
454 × 430	195220	216	200	93%
549 × 515	282735	216	200	93%
760 × 725	551000	216	184	85%
841 × 825	693825	216	160	74%

The data augmentation stage involves the acquisition of new and robust data by applying artificial changes for enhancing the quality of the original data [20]. The muzzle pattern data was applied with various techniques, such as affine transformation and noise injection, considering various factors that influenced the data collection process (e.g., shooting angle, light intensity, perspective, noise, etc.) to increase the robustness of the data and object recognition rate. This process facilitates the detection of objects in the image more easily even if they are small or located in a different position within the image. Next, the muzzle pattern image data was proportionally resized using bilinear interpolation. Bilinear interpolation is a method for obtaining the pixel value of the image and computes the linear sum of the values of four pixels surrounding the adjacent real coordinates in the data multiplied by the weight. This method was used to resize 2 × 2 neighboring pixels to 200 × 200. As shown in Fig. 3, the original and comparison data were resized to have the same size and shape for the matching process [21].

Figure 3 Convert images using bilinear interpolation

In the next step, CLAHE was applied to compensate for contrast differences that can reduce the object recognition rate while setting a contrast limit to prevent extreme noise amplification [3]. CLAHE divides each image into smaller areas, and increases the contrast in dark areas and lowers the contrast in bright areas, thereby equalizing the local contrast within the image. Then, the unique features, or keypoints, of the muzzle patterns were extracted through keypoint detection and generation of feature descriptors that contain vector-based information describing the keypoints [19]. This feature extraction step involved the following procedures. First, the images were smoothed using the Gaussian kernel to prevent high frequency noise from affecting the feature extraction process. Then, the muzzle pattern data were screened by setting the ROI of the data and applying the Harris Corner Detector. The Harris Corner Detector infers the keypoints of an image by measuring the amount of change occurring in the pixel values when the ROI is shifted. The ROIs where pixels' intensity variation is significant when shifted in vertical, horizontal, or diagonal directions were determined as corners. Only the corners that show variations of 10% or more of the maximum value were recognized and marked as keypoints. Next, the binary descriptors of the keypoints were extracted as shown in Fig. 4. Two pixels for pair-wise comparison were selected from the different parts of an image patch of a keypoint, and their sizes were compared and converted into binary values (either 0 or 1). In this way, a total of 128 pairs of binary values were compared in the image patch for keypoint (k) to generate a 128-bit binary descriptor for the respective keypoint.

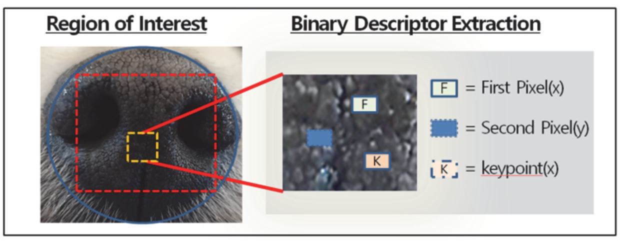

Figure 4 Binary descriptor extraction

In the muzzle pattern data matching step, it was checked whether the source and stored muzzle patterns were recognized as the same using the extracted keypoints. The extracted keypoints data were saved in the database through an application programming interface (API), then the keypoints data of the 200 muzzle patterns were processed using the ORB algorithm applied with BEBLID to match the muzzle patterns of the source and stored data. In doing so, the scales of the descriptors were adjusted from 0.25 to 2 to find the optimal BEBLID scale coefficient. It is possible for an object matching and recognition algorithm to contain outliers that increases the false match rate and reduces accuracy. To prevent this, RANSAC was used to remove any outliers. RANSAC estimates the mediating variable of a mathematical model from a randomly selected sample of an observation data that includes outliers and then finds the model with the highest internal value through iteration [3].

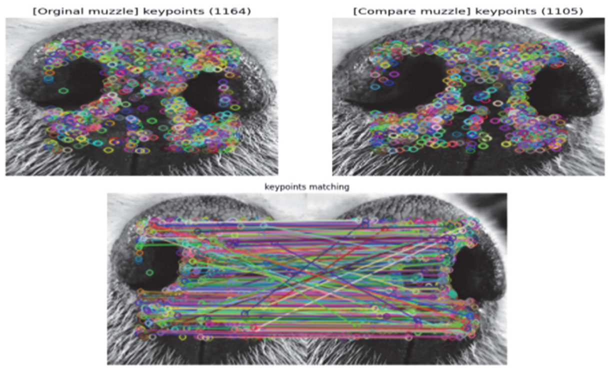

Figure 5 Keypoints matching between the source and stored data

The duplicate matching removal step checks the redundancy of the matched data to ensure that the binary descriptors were processed separately from other classifications. The dog muzzle patterns in the muzzle pattern database were managed using SQL and DataGrip tools. Lastly, the final results were analyzed to confirm the matching accuracy by performing image matching operations, such as perspective-n-point, homography estimation, image stitching, plane tracking, and real-time pose estimation for each corresponding keypoint.

3.2 Experimental Set-up

The experiment was conducted by comparing the source data taken with a smartphone camera with 200 data stored in the database. The experimental set-up was configured based on Open cv-contrib-python version 4.5.5.64 as illustrated in Fig. 6. An internal virtual machine manager (VNM) was installed using CentOS7 as its operating system. The OpenCV-Python consisted of Flask (Python micro-web framework), Nest.js (node express-based web framework), and an image processing package module. The database was set up using PostgreSQL. Data Grip (a database connection client tool), Python IDE Pycharm Professional, Web storm (a web development IDE), and Docker (for container creation). For muzzle pattern recognition, a REST API environment was configured for extracting the features of the muzzle patterns and storing, matching, registering, modifying, and deleting the data, and performance and functional experiments were conducted.

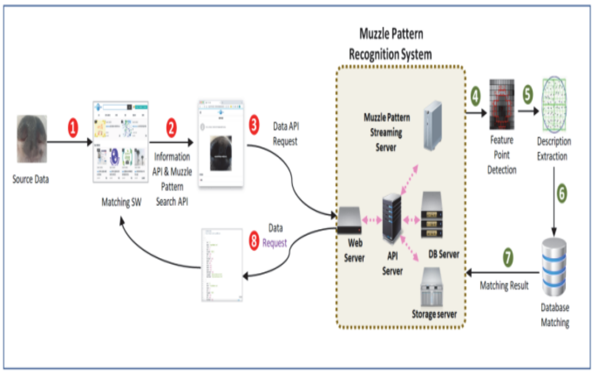

Figure 6 Experimental set-up

The nearest neighbor matching method using the Hemming distance algorithm, which checks the overall similarity while matching the feature descriptors of the images one by one, was adopted [9]. Also, to examine the data search and matching A software was developed to assist with data search and examine the matching performance. The cross-check method was used to confirm the images as a match. In other words, the features were considered to be matching if a feature descriptor of image A was matched with a feature descriptor of image B, and if the match could be confirmed in the opposite direction as well, from image B to image A [22].

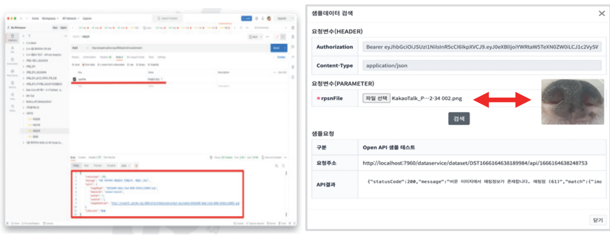

Figure 7 Image cross check matching

4 RESULTS AND DISCUSSION

The experiment presented in this study processed and matched a dataset of 200 dog muzzle patterns. The experiment was performed in two parts. First, the features extracted from the muzzle patterns of the source and stored data were compared using the ORB algorithm applied with the BEBLID descriptor (scale coefficient set as K = 1). Second, the same experiment was performed again while changing the value of the BEBLID scale coefficient to determine the optimal value.

4.1 Key Findings from Applying BEBLID to the ORB Algorithm

The ORB algorithm applies the Harris Corner Detector to extract the most useful keypoints. Even if the size of the image is changed or the image is rotated, ORB is capable of extracting accurate keypoints. The muzzle pattern feature points were extracted and stored in the database through pre-processing based on signal processing, and the recognition rate was investigated by comparing and analyzing the feature points of the muzzle pattern previously stored in the database.

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

    # ORB & BEBLID process
    kp_detector = cv2.ORB_create(10000)
    keypoint = kp_detector.detect(tempImg, None)
    # desc_detector = cv2.ORB_create()
    desc_detector = cv2.xfeatures2d.BEBLID_create(0.75)
    _, descriptor = desc_detector.compute(tempImg, keypoint)
    return keypoint, descriptor
```

Figure 8 Descriptor value processing

The original image and the comparative image in the parameter returned the keypoint using the descriptor of the image in the Keypoint_process (img, name) module.

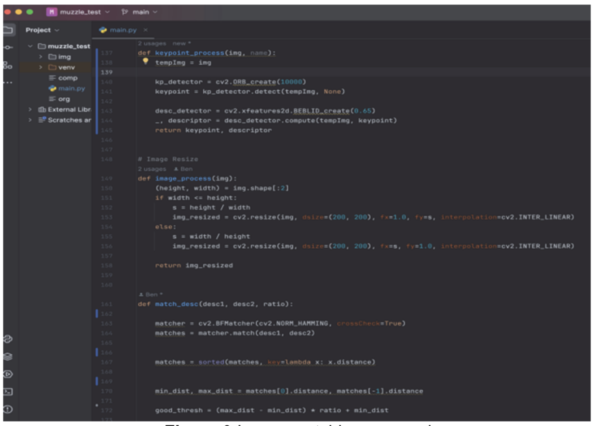

Figure 9 Image matching processing

In BEDBLID, the recognition rate varies greatly as the scale factor value is added, which is the default setting for creating an image pyramid by listing one image as multiple images with different reduction ratios, and then making the same feature part of the image pyramid into a feature descriptor. Each step is transformed into a scale factor value to match the characteristics of the image without being affected by color, rotation, and scale. When ORB was applied to the source data, the algorithm extracted 1161 keypoints, whereas the stored data contained 1,105 keypoints. The two data shared 413 matching points, and the ORB algorithm matched 276, representing a 67% matching rate, as shown in Fig. 10.

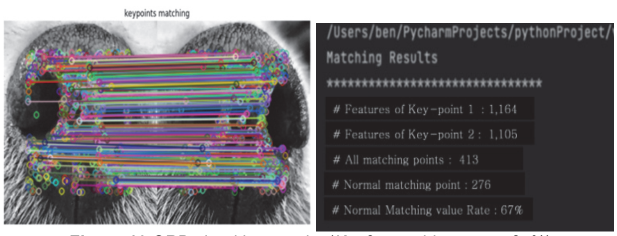

Figure 10 ORB algorithm results (K = 0, matching rate = 67%)

BEBLID, on the other hand, performs the process of comparing each descriptor of the first image with all of the descriptors of the second image to check the keypoints matching between the two images. The comparison of descriptors is conducted based on the Hamming distance technique, which computes the difference in the number of bits between each descriptor pair to check whether they match. At this time, the K value of BEBLID was set as 1. When the BEBLID value was applied to the two data compared using the ORB algorithm, 337 matching points were recognized, improving the matching rate to 76% as shown in Fig 11.

Figure 11 BEBLID results (K = 1, matching rate = 76%)

Tab. 3 presents the results obtained using the ORB algorithm and after applying BEBLID to the ORB algorithm. By applying BEBLID, it was possible to increase the number of matching points by 61 and the matching ratio by 9%.

Table3 Muzzle pattern matching results

Items	Keypoint 1 (Orginal)	Keypoint 2 (Compare)	Matches	Inliers	Matching rate
ORB Descriptor (k = 0)	1,164	1,105	413	276	66.8%
BEBLID Descriptor (k = 1)	1,164	1,105	442	337	76.2%

4.2 Deriving the Optimal BEBLID Scale Coefficient

To derive the optimal value of BEBLID scale coefficient K, the changes in matching accuracy were examined as the BEBLID's K value was adjusted from 0.25 to 2. The matching rate achieved by BEBLID using variant K values for the 200 source and stored data (containing 1150 and 1498 keypoints, respectively, according to the ORB algorithm) was tested by matching identical data 10 times and non-identical 10 times. The results of this experiment are presented in Tab. 4. As mentioned above, BEBLID descriptor's scale factor, the major variable in the experiment, refers to the image pyramid, which can have various versions of the same image in different resolutions, with each level depending on the scale of the original image. The scale factor determines the relative scale between two successive pyramid levels. For instance, a scale factor of 2.0 results in image pyramids that are half the resolution of the previous version. In this way, the performance and speed of keypoint detection can be adjusted by changing the scale factor of the image pyramid. While a lower scale factor produces more accurate results, it can be more costly to compute, which is why it is important to derive the most suitable value. Keypoint extraction was performed using SIFT, SURF, BRISK, ORB, and three descriptors, BRIEF, and BEBLID were computed for each keypoint. The scale factor of the BEBLID descriptor was variably applied between 0.25 to 2.0 to test the performance and speed of keypoint extraction and matching. BEBLID descriptor's scale factor refers to the scaling factor of the image pyramid, and scale factor determines the relative scale between two consecutive pyramid levels. A value of 2.0 means that each pyramid level will have half the resolution of the previous level. Values above 2.0 were not included because they would result in erroneous data due to the differences in sizes and resolutions being compared. The highest matching rate was obtained when the BEBLID's K value was set as 0.75 in the ORB algorithm. When K was 1, the matching rate was 76%, but when K was adjusted to 0.75, the matching rate rose to 87%. The results also show that, while the K value does not influence the results significantly when the objects are large, it can bring a notable change in the matching rate when the objects are small, like muzzle patterns. It should be noted that the lowest value of K was set as 0.75 because the matching process takes a relatively longer time without any changes in the matching rate when applying a smaller value.

Table 4 Deriving BEBLID's optimal K value

Items	Original Data	Compare Data	Descriptor	Factor	Matches	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%
				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%
				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%

As a result of comparing the BEBLID variable descriptor value by algorithm, the matching rate was the highest at the variable value of 0.75 in ORB. There showed a difference in the matching rate depending on the size of the image data, and the matching rate can be improved as the variable descriptor value. Adjusting the value of the BEBLID descriptor in binary-based feature point extraction resulted in an increase in the matching rate.

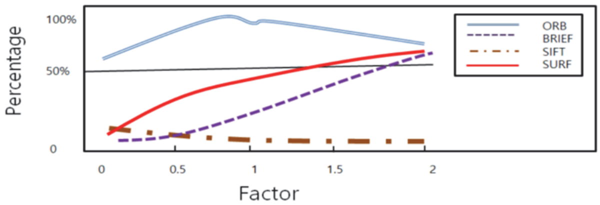

Figure 12 Matching rate graph

A one sample *T*-test was performed to validate the data, which resulted in a *p*-value smaller than 0.001, demonstrating the high significance of the results obtained. In conclusion, extracting feature points with ORB and calculating descriptor with BEBLID provided the best matching performance.

Table 5 One sample *T*-test verification

Items	Test Value = 0					
	t	*df*	Sig. (*p*-value)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
SIFT	37.000	11	.000	9.250	8.70	9.80
SURF	14.588	11	.000	45.333	38.49	52.17
BRISK	4.930	11	.000	22.333	12.36	32.30
ORB	19.045	11	.000	72.083	63.75	80.41

5 CONCLUSIONS

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.

Acknowledgments

This research was financially supported by Hansung University.

6 REFERENCES

[1] Liu, S. & Silverman, M. (2001). A Practical Guide to Biometric Security Technology. *IT Professional, 3*(1), 27-32. https://doi.org/10.1109/6294.899930

[2] Kim, Y., Park, W., Roh, M.-C., & Shin, J. (2020). Group face: Learning Latent Groups and Constructing Group-based Representations for Face Recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5620-5629. https://doi.org/10.1109/CVPR42600.2020.00566

[3] Ng, R., Tay, Y., & Mok, K. M. (2008). A review of iris recognition algorithms. *Proceedings of the 2008 International Symposium on Information Technology, 2*, 1-7. https://doi.org/10.1109/ITSIM.2008.4631656

[4] Shim, Y. B. & Park, H. J. (2015). A Study on a Violence Recognition System with CCTV. *Journal of the Digital Contents Society, 16*(1), 25-32. https://doi.org/10.9728/dcs.2015.16.1.25

[5] Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision, 60*(2), 91-110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[6] Rublee, E., Rabaud, V., & Konolige, K. (2011). ORB: an efficient alternative to SIFT or SURF. *IEEE International Conference*, 2564-2571. https://doi.org/10.1109/ICCV.2011.6126544

[7] Tareen, S. & Saleem, Z. (2018). A Comparative Analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. *Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 1-10. https://doi.org/10.1109/ICOMET.2018.8346440

[8] Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). BRIEF: Binary robust independent elementary features. *Computer Vision - ECCV 2010*, 778-792. https://doi.org/10.1007/978-3-642-15561-1_56

[9] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *International Conference on Computer Vision*. 2564-2571. https://doi.org/10.1109/ICCV.2011.6126544

[10] Suárez, I., Sfeir, G., Buenaposada, J., & Baumela, M. (2020). BEBLID: Boosted efficient binary local image descriptor. *Pattern Recognition Letters, 169*, 366-372. https://doi.org/10.1016/j.patrec.2020.04.005

[11] Fischler, C. & Bolles, M. (1981). A paradigm for model fitting with applications to image analysis and automated cartograph. *Random sample consensus. ACM,* 381-395. https://doi.org/10.1145/358669.358692
[12] 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. Jpn. Soc. Agric. Inform,* 596-600.
[13] Barry, B., Gonzales-Barron, U., McDonnell, K., Butler, F., & Ward, S. (2007). Using Muzzle Pattern Recognition as a Biometric Approach for Cattle Identification. *Trans. ASABE*, 1073-1080. https://doi.org/10.13031/2013.23121
[14] Noviyanto, A. & Arymurthy, A. (2013). Beef cattle identification based on muzzle pattern using a matching refinement technique in the SIFT method. *Comput. Electron. Agric.* 77-84. https://doi.org/10.1016/j.compag.2013.09.002
[15] Kumar, S. (2018). Deep learning framework for recognition of cattle using muzzle point image pattern. *Measurement, 116*, 1-17. https://doi.org/10.1016/j.measurement.2017.10.064
[16] Tharwat, A., Gaber, T., Hassanien, A., Hassanien, H., & Tolba, M. (2023). Cattle Identification Using Muzzle Print Images Based on Texture Features Approach. *The 6th International Conference on Electrical Engineering, Control And Robotics*, 853.
[17] Tong, H., Li, M., Zhang, H., & Zhang, C. (2004). Blur detection for digital images using wavelet transform. *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME), 1*, 17-20.
[18] Kelleher, J. & Tierney, B. (2019). Data Science. *The MIT Press Essential Knowledge Series*.
[19] Liu, T., Wu, D., & Wang, L.Y. (2020). Development process of animal image recognition technology and its application in modern cow and pig industry. *IOP Conf. Ser.: Earth Environ. Sci.* https://doi.org/10.1088/1755-1315/512/1/012090
[20] Ross, A. & Prabhakar, S. (2004). An Introduction to Biometric Recognition Jain. *IEEE Trans. Circuits Syst. Video Technol, 2*, 1-7.
[21] Zuiderveld, K. (1994). Contrast Limited Adaptive Histogram Equalization. *Graphics Gems*, 474-485. https://doi.org/10.1016/B978-0-12-336156-1.50061-6
[22] Khan, M., Ishaq, M., Swain, M., & Kwon, S. (2023). Advanced Sequence Learning Approaches for Emotion Recognition Using Speech Signals. *Intelligent Multimedia Signal Processing for Smart Ecosystems*, 307-325. https://doi.org/10.1007/978-3-031-34873-0_13

Contact information:

Jeon Seong YANG, PhD Candidate
Dept. of Knowledge Service & Consulting, Hansung University,
116, Samseongyo-ro 16-gil, Seongbuk-gu, Seoul, Republic of Korea
E-mail: jerry27@ hansung.ac.kr

Seok Kee LEE, Professor
(Corresponding author)
School of Computer Engineering, Hansung University,
116, Samseongyo-ro 16-gil, Seongbuk-gu, Seoul, Republic of Korea
E-mail: seelee@hansung.ac.kr