2018-2019
Coursework submission
Cover sheet

Student name……

College……

CRSID……

Submit to Graduate Education
Date of submission

Module name

Module Code No.

Assignment No.

Assignment mark/grade

Assessor comments

GE Records……..

GE stamp date and initials

CRSID……

Module No……

Assignment No……

Automatic Identification of Manta Rays

Zhuoni Jie
Department of Computer Science and Technology,
University of Cambridge
zj245@cam.ac.uk

***Abstract*— Visual identification of individual animals that bear unique natural body markings is an important task in wildlife conservation. As the photo databases of animal markings grow larger, scientists are often faced with the task of matching observations against databases of hundreds or thousands of images. Existing photo-identification solutions have constraints on image quality and appearance of the pattern of interest in the image. These constraints limit the use of photos from citizen scientists. As complementation of traditional keypoint feature based methods, convolutional neural networks (CNN) have shown promising results in many computer vision tasks. However, there is limited study about the comparison of performance of keypoint features with CNNs for descriptor matching and individual identification. In this project I compare three methods - InceptionV3 CNN, deep learned DELF keypoint features, and SIFT features - for visual identification of manta rays based on unique natural markings that is robust to occlusions, viewpoint and illumination changes. For InceptionV3, I adapt methods developed for face re-identification and implement a deep CNN to learn features from its layers. For DELF features, I use a pre-trained model trained on Google landmarks datasets. For SIFT features, I implement by using OpenCV library. I evaluate the proposed methods on image databases of manta ray belly patterns, and compared their qualitative and quantitative performances. DELF shows its good performance in identifying manta rays, while SIFT performs much faster than CNN based methods while also maintaining nice performance results.**

I. INTRODUCTION

The identification of individuals of a particular species is a vital requirement for ecological research and conservation [1]. Visual identification, which is an increasingly popular technique to study animals by using their unique and sufficiently stable natural markings in photo databases, is effective and non-invasive for distinguishing animal individuals [2]. Tracking population dynamics of animals such as manta rays is critical owing to their vulnerable conservation status, and economic importance in both ecotourism and fisheries [3]. Manta rays are born with natural spot patterning that stays unchanged throughout their life and can be used like human fingerprints to identify individuals [4]. These individual patterns enable scientists to study populations, track long range movements and record habitat use. Examples of manta ray spot patterns are shown in Fig. 1.

While good accuracies have been achieved in automated photographic identification systems for many species observed on land or at the surface of the sea [5][6], recognition of animals underwater poses significant additional challenges. Specifically, the task of recognizing manta rays

Fig. 1. Examples of manta ray spot patterns.

is challenging because of the heterogeneity of photographic conditions and equipment used in acquiring manta ray photos [7], the generally low contrast and high variation in size and shape exhibited by their characteristic ventrsal body markings, and the region of interest's (ROI) high flexibility resulting in wide variation from small regular spots to frayed blotches [4]. Other factors include poor visibility for underwater images, illumination, and small objects occluding the pattern on the animal [8].

Local descriptors based on orientation histograms, such as the Scale-Invariant Feature Transform (SIFT) [9] and Speeded Up Robust Features (SURF) [10], have been not only successful in recognition but also in descriptor matching. The current state-of-the-art manta ray recognition system Manta Matcher [4] is based on automated extraction and matching of keypoint features using the SIFT algorithm with relative modifications and enhancements. Manta Matcher requires the user to manually align and normalize the 2D orientation of the manta ray within the image, and select a rectangular region of interest containing the spot pattern. The Manta Matcher works best with photos taken perpendicular to the manta's ventral pattern with no reflective particles in the water and in good lighting conditions [8]. In practice, these constraints limit the use of photos, and some marine biologists still prefer to use a decision tree that they run manually [8].

Convolutional neural networks (CNN) have been applied to the task of animal identification as a classification problem [11][12][13]. Many results indicate that features learned via convolutional neural networks outperform previous descriptors on classification tasks by a large margin, and it has been shown that these networks trained on ImageNet also perform very well on other datasets or recognition tasks different from those they were trained on [14][15]. Still, there is limited study about the comparison of performance of

keypoint features with convolutional networks for descriptor matching and individual identification.

In this work I focus on comparing the use of deep learning features versus keypoint features in manta ray identification task. By investigating these different methods, I aim to develop robust algorithms for recognizing manta spot patterns and potentially eliminating some constraints of previous wildlife matching systems such as requirements for high image quality and a clear view of the animal markings in the image.

The main contribution of this work is a study of visual manta ray identification techniques. I compare (1) features extracted from various layers of CNNs, (2) a CNN-based local feature with attention, DELF (DEep Local Feature) [31], with (3) standard SIFT descriptors. By quantitatively and qualitatively evaluating the features, I investigate different techniques' robustness to viewpoint changes, small occlusions and lighting conditions, and therefore ability to match images from citizen scientists who produce images with a wide range of conditions.

The paper is organized as follows. Section II reviews related work on identification tasks using different methods. Section III presents methods overview used in this project. Section IV describes the design and implementation of my experiments, and provides evaluation results. Section V concludes the paper and provides future directions.

II. RELATED WORK

The techniques that have been proposed for identification of animal natural markings vary in the core methods used, amount of user involvement, and ability to be adapted to different species.

Some matching tasks has been approached by exhaustively generating two-dimensional affine transformations based on user provided key points and comparing each transformation of a candidate example with the examples stored in a repository [16][17]. The algorithm was implemented in a solution called APHIS (Automated Photo-Identification Suite). However, the method requires a user to select key points and identify the most distinctive spots for each image.

Some methods have been developed for specific species and are not easily transferable to other species. For example, high-contrast colour patterns of humpback whale flukes [18] and dolphin dorsal fins [19] are matched by extracting hand-crafted features from corresponding segments obtained by overlaying a grid on a region of interest. This method is not robust to viewpoint changes.

Two current systems for animal identification used in practice (Manta Matcher [4], HotSpotter [20]) are based on automated extraction and matching of SIFT keypoint features with different modifications and enhancements to work on specific cases. While the SIFT algorithm works well on images that clearly show the pattern of interest, it is not robust to large changes in camera viewpoint, occlusions and variations in illumination.

The task of animal visual identification can also be related to the face recognition problem that has been extensively studied with deep learning in recent years [21][22]. The main idea is to use CNN for learning embeddings in a feature space where matching patches are closer to each other than non-matching patches. However, they do not learn keypoint detection and their resprctive descriptors. In the past few years, several global descriptors based on CNNs have been proposed to use pretrained [23][24] or learned networks [25]. CNNs have also been used to detect, represent and compare local image features. Verdie et al. [26] learned a regressor for repeatable keypoint detection. Yi et al. [27] proposed a generic CNN-based technique to estimate the canonical orientation of a local feature and successfully deployed it to several different descriptors. MatchNet [28] and DeepCompare [29] have been proposed to jointly learn patch representations and associated metrics. Recently, LIFT [30] proposed an end-to-end framework to detect keypoints, estimate orientation, and compute descriptors. However, these techniques are not designed to learn to select semantically meaningful features. Noh et al. [31] proposed DELF, an attentive local feature descriptor to identify semantically useful local features for image retrieval.

III. METHOD OVERVIEW

The goal of this task is to learn features in manta ray images and perform identification of individuals. I achieve this by using and comparing three methods. The first is to use fine-tuning InceptionV3 [32] based model, the second is to use CNN-based local keypoint features, and the third is to use SIFT keypoint features.

A. Fine-tuning InceptionV3 Based Model

Within the object detection and recognition literature, the detection of salient regions with deep architectures has been much discussed. In [33], features in later layers were shown to correspond to fine details in the receptive fields covered by those features. Here I use Google's pre-trained Inception Convolutional Neural Network [32] to perform image recognition and train the last layer of CNN, which distinguishs the specific, higher-level features of each class relative to this manta ray identification task, using back-propagation on the manta ray dataset. As an alternative, I also took the output of the intermediate layer prior to the fully connected layers as features (bottleneck features) and trained a linear classifier - Support Vector Machine - on top of it. This generates similar results as the fine-tuning approach.

The original Inception V3 model has achieved 78.0% top-1 and 93.9% top-5 accuracy on the ImageNet test dataset containing 1000 image classes. InceptionV3 gets good performance by using a 48-layer deep architecture, incorporating inception modules, and training on 1.2 million images. The inception modules take several convolutional kernels of different sizes and stack their outputs along the depth dimension in order to capture features at different scales. InceptionV3 improved utilization of the computing resources inside the network by using a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant.

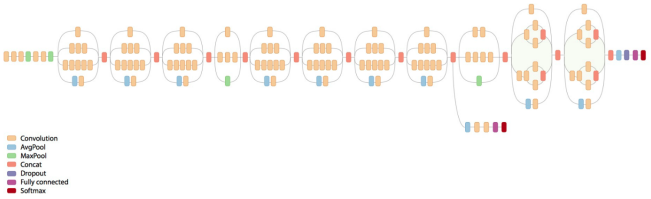
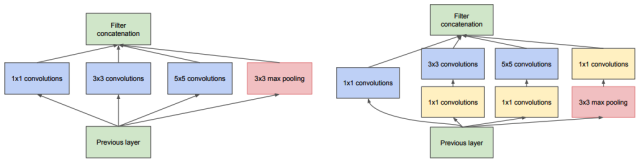


Fig. 2. Basic Inception CNN architecture.



(a) Inception module, naïve version (b) Inception module with dimension reductions

Fig. 3. Inception module.

B. Dense Localized Feature (DELF) Extraction

The DELF extraction method can be decomposed into six main stages: (1) image preprocessing, (2) image enhancement, (3) dense localized feature extraction, (4) keypoint selection, (5) dimensionality reduction, and (6) feature matching.

As performed in [4], image preprocessing is a simple alignment process in which the user normalizes the 2D orientation of the ray within the image and selects a rectangular ROI containing the spot pattern, which is currently done manually. The image used here is automatically enhanced through noise removal and adaptive contrast equalization.

As for DELF features, I use features extracted using the pre-trained Google-landmarks model as used in this paper [34]. In this model, dense features are extracted from an image by applying a fully convolutional network (FCN), which is constructed by using the feature extraction layers of a CNN trained with a classification loss. The original paper employs an FCN taken from the ResNet50 [35] model, using the output of the conv4_x convolutional block. To handle scale changes, the original algorithm explicitly constructs an image pyramid and apply the FCN for each level independently. The obtained feature maps are regarded as a dense grid of local descriptors. Features are localized based on their receptive fields, which can be computed by considering the configuration of convolutional and pooling layers of the FCN. The algorithm uses the pixel coordinates of the center of the receptive field as the feature location. The receptive field size for the image at the original scale is 291×291. Using the image pyramid, features that describe image regions of different sizes are obtained.

Unlike traditional keypoint detectors, keypoint selection in DELF comes after descriptor extraction. A technique is designed to effectively select a subset of the features by training a landmark classifier with attention to explicitly measure relevance scores for local feature descriptors. This model not only encodes higher level semantics in the feature map, but also learns to select discriminative features for the classification task. Moreover, similar to CNN-based image classification techniques, the DELF model implicitly learns to learn invariances to pose and viewpoint.

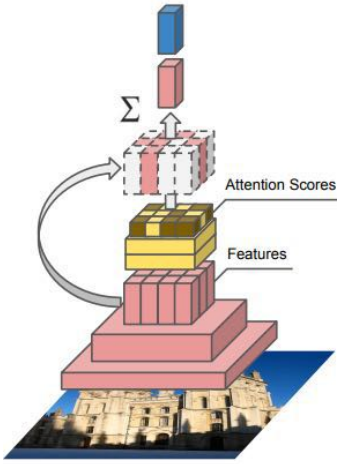


Fig. 4. The network architecture used for attantion-based keypoint selection training in DELF.

The selected features are L_2 normalized, and their dimensionality is reduced to 40 by Principal Component Analysis (PCA), which presents a good trade-off between compactness and discriminativeness. Then the features once again undergo L_2 normalization.

The features are matched based on nearest neighbor search, which is implemented by a combination of KD-tree [36] and Product Quantization [37]. When a matching is performed, the model performs approximate nearest neighbor search for each local descriptor extracted from the query image. Geometric verification is performed using RANSAC [38], and the number of inliers are employed as the score for the other retrieved image.

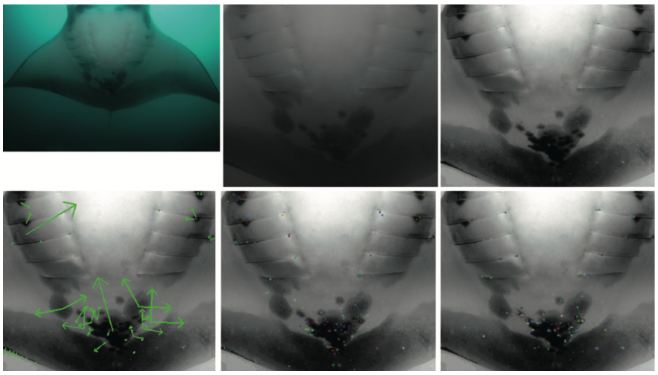


Fig. 5. Top row, left to right: Original image; grayscale candidate region; enhanced image after noise filtering and contrast adjustment. Bottom row: Visualization of features extracted using SIFT (left), SURF (middle), and ORB (right). [4]

C. Scale-Invariant Feature Transform (SIFT) Extraction

Similar to DELF method, the SIFT extraction method can also be decomposed into five main stages: (1) image prepro-

cessing, (2) image enhancement, (3) data augmentation, (4) SIFT feature extraction, (5) feature matching.

The image preprocessing and image enhancement were done as described in the above section. For SIFT feature extraction and matching, I used data augmentation. The main challenge with this manta ray dataset is the small number of images per individual with over two-thirds of the manta rays having two or three sightings. Data augmentation is an automatic way to boost the number of different images used to train models. Here I use *ImageDataGenerator* function from Keras library to do random rotation/reflection, flip, and shift of the images.

The SIFT algorithm detects distinctive features at keypoints in the image, and then represents those features in terms of a parametric description of the local image variation in the vicinity of the keypoints at a carefully chosen scale of analysis. SIFT algorithm uses Difference of Gaussians (DoG) which is an approximation of Laplacian of Gaussian. Once this DoG are found, images are searched for local extrema over scale and space to detect potential keypoints. The model then uses Taylor series expansion of scale space to get more accurate location of extrema, and if the intensity at this extrema is less than a threshold value (0.03 as per the paper [9]), it is rejected. Low-contrast keypoints and edge keypoints are eliminated. Then an orientation is assigned to each keypoint to achieve invariance to image rotation. A neighborhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction is calculated in that region. After a keypoint descriptor is created, the 16×16 neighbourhood around the keypoint is taken. It is devided into 16 sub-blocks of 4×4 size. For each sub-block, 8 bin orientation histogram is created. Therefore, a total of 128 bin values are available. It is represented as a vector to form keypoint descriptor.

The SIFT algorithm was chosen due to its ability to extract and match features in a way which is robust to changes in size and 2D rotation, and also resilient to changes in 3D viewpoint, addition of noise, and change in illumination. In terms of the characteristic patterns present on the ventral surface of manta rays, SIFT keypoints are typically localized at significant spots and other markings. Information on the shape, contrast, and dominant orientation of markings is represented by the feature descriptors.

FLANN (Fast Library for Approximate Nearest Neighbors) method was chosen for feature matching. It contains a collection of algorithms optimized for fast nearest neighbor search in large datasets and for high dimensional features. For FLANN based matcher, two dictionaries need to be passed which specify the algorithm to be used, its related parameters etc. It works faster than the Brute-Force Matching method for large datasets, in which one feature in first set is matched with all other features in second set and returns the closest one. Ratio test is employed to discard many of the false matches arising from background clutter [9].

Fig. 5 illustrates image enhancement and features extraction using an example manta ray image.

IV. EXPERIMENTS

A. Datasets

I perform experiments on a dataset of 720 images of 265 different manta rays taken under widely different conditions as used in [4]. Over 80% manta ray individuals have a composition from 0.3% to 0.5% in this dataset, which means most individuals have 2 or 3 photos, denoting this is a relatively balanced dataset. The images were taken by members and associates of the Manta Ray & Whale Shark Research Centre, Marine Megafauna Foundation, Tofo Beach, Inhambane, Mozambique. Most of the images (581 photos of 214 individuals) depict reef manta rays *M. alfredi* [40], and the remainder (139 photos of 51 individuals) depict giant manta rays *M. birostris*. Of the 214 reef mantas, 161 were visually assessed as being female and 51 as being male (there were two rays for which sex could not be determined), and of the 51 giant mantas, 38 were visually assessed as being female and 6 as being male (with the remaining 7 being indeterminable).

B. Implementation Details

1) InceptionV3 Based Model Feature Extraction and Identification: I first downloaded the most up-to-date InceptionV3 model via Tensorflow's repository [41] with the parameters learned through training on the ImageNet dataset. I truncated the last layer before predictions of the pre-trained network, the 'PreLogits' layer, and replaced it with a new softmax layer relevant to the task. The size of this layer is [None, 1, 1, 2048], so there are 2048 filters each with size 1×1. This layer is created as a result of applying an average pooling operation with an 8×8 kernel to the Mixed layers and then applying dropout. I changed this layer to a fully connected layer by eliminating (squeezing) the two dimensions that are 1, leaving with a fully connected layer with 2048 neurons. I then created a new output layer that takes the prelogits as input with the number of neurons corresponding to the number of classes. To train only a single layer, I specified the list of trainable variables in the training operation. What the network is learning during training is the connection weights between the 2048 neurons in the prelogits layer and the 10 output neurons as well as the bias for each of the output neurons. I also applied a softmax activation function to the logits returned by the network in order to calculate probabilities for each class. After isolating the single layer to train, the rest of the code is fairly straightforward and typical for a neural network used for classifcation. Average cross entropy was used as a loss function and the Adam Optimization function was used with a learning rate of 0.01. The accuracy function measured top-k accuracy. Finally, I used an initializer and a saver so as to save the model during training and restore it at a later time.

During training, I passed batches of images to the constructed network. I trained using early stopping, which is one method for reducing overfitting on the training set. Early stopping requires periodically testing the network on a validation set to assess the score on the cost function

(average cross entropy here). If the loss does not decrease for a specified number of epochs, training is halted. In order to retain the optimal model, each time the loss improves, the model will be saved. Then, at the very end of training, the model that achieved the best loss on the validation set will be saved. I implemented a type of early stopping by using a single validation set and stop training if the loss does not improve for 20 epochs.

The activation maps in the neural network can both classify the image and localize class-specific image regions, and I visualized one activation map as shown in Fig. 6. I also visualized different feature extractors (filters) emerge at different layers during the training of the network. Low layer features usually include lines, contrast, colors. Medium layer features usually include corners or other edge/color conjunctions and textures. High layer features usually include more complex and class specific features. From Fig. 7 we can see that the deeper we dive into the network, the more complex the filters are.

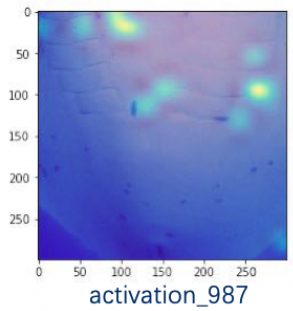


Fig. 6. Visualization of the activation map.

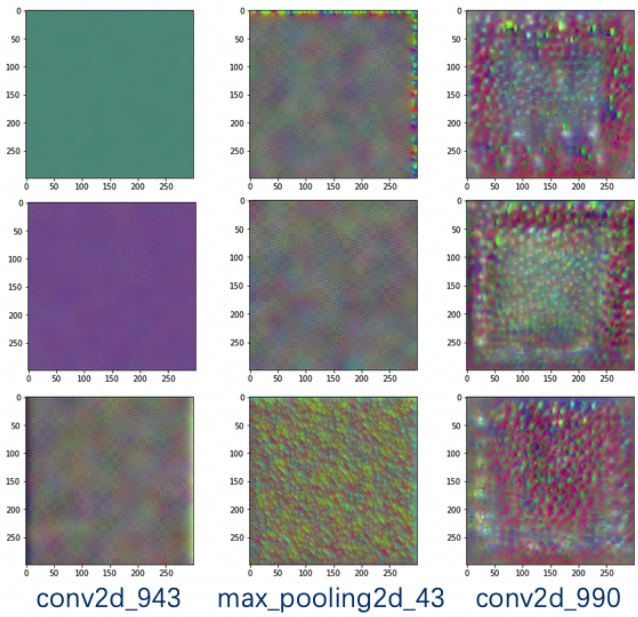


Fig. 7. Visualization of the filters in the network.

2) DELF Feature Extraction and Matching: For DELF feature extraction, I use the trained DELF model. The original model employed landmarks dataset [] for fine-tuning descriptoes and training keypoint selection. The "full" version part of the dataset was used to train the attention model, while the "clean" version part of the dataset obtained by a SIFT-based matching procedure was employed to fine-tune the network for image tasks. The model identifies the top K (= 60) nearest neighbors for each feature in a query and extract up to 1000 local features from each imageeach feature is 40-dimensional. A sample image matching using DELF features is as shown in Fig. 11 and Fig. 13.

3) SIFT Feature Extraction and Matching: My implementation makes use of the OpenCV 3.4.2 Computer Vision library [39]. For FLANN parameters, I used FLANN_INDEX_KDTREE algorithm with *trees = 5*, *checks = 50*, and *k = 2*. For ratio test, I used *ratio = 0.75*.

C. Evaluation and Discussion

The confusion matrix for the original dataset's top-1 InceptionV3 model accuracy is shown in Fig. 8, and we can find there are a number of correct matches as well as mismatches. In practice, it is already very helpful for scientists to get correct identification result in top-k (k can be as large as 25) matching results. Therefore, I further investigate top-k accuracy results when using different methods and varied augmentation methods.

To aim for better identification performance, I did data augmentation using random rotation/reflection, flipped, and shift augmentation, expanding the dataset to 2160 images, and got the following identification results as shown in Fig. 9 and Fig. 10. I compare the identification top-k (k = 1, 5, 10, 25, 50, 75, 100) accuracy results using three methods (InceptionV3, DELF, and SIFT), and with different data augmentation amount (no augment, 720 augment, 1440 flipped/shift augment, and 2160 augment).

From the results we can see deep-learned DELF feature beat the other techniques in all the top-k scores. CNN based feature methods, extracting features from layers in InceptionV3 model, have performances better than other keypoint methods in top-1 accuracy, but fall behind in top-k accuracy when k is larger than 5. We can also see different data augmentation methods have influences on CNN model performance, with flipping generating better results than shift augmentation. This is probably because shifting can sometimes lose areas of interests which are close to image edges. For the traditional SIFT keypoint descriptors, we can see although SIFT get relatively lower top-1 accuracy scores, with different data augmentation, SIFT generate very nice performances after top-5 accuracy. All the keypoint features (traditional SIFT with CNN-based DELF) reach accuracy very close to 100% after top-25 accuracy.

Notice that there are some images that have no match in the database, and there are some pictures having different degrees of occlusion. All of these make this dataset more challenging.

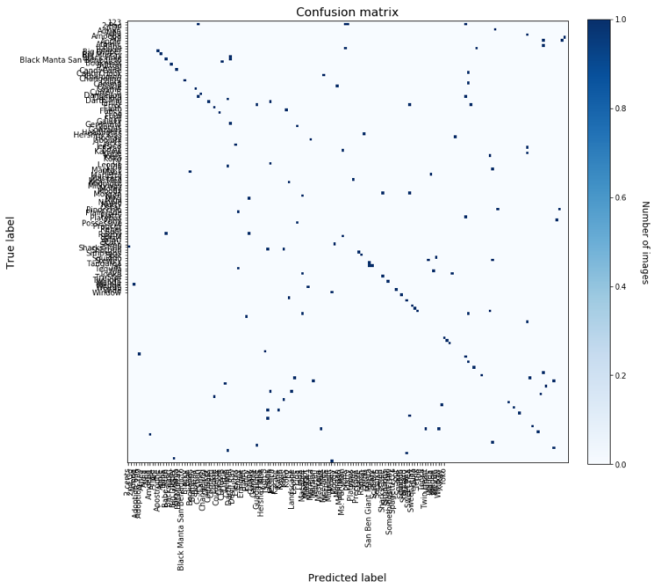


Fig. 8. Confusion matrix of top-1 accuracy for the fine-tuned InceptionV3 model.

	Top 1 (%)	Top 5 (%)	Top 10 (%)	Top 25 (%)	Top 50 (%)	Top 75 (%)	Top 100 (%)
InceptionV3 flipped aug	36.68	43.18	55.68	72.72	84.09	86.36	88.63
InceptionV3 shift aug	23.81	40.26	50.65	68.83	79.22	83.12	85.28
InceptionV3 no aug	27.27	38.63	56.82	77.27	88.64	88.63	88.63
DELF no gug	54.23	99.23	100.00	100.00	100.00	100.00	100.00
SIFT no aug	37.34	96.43	99.68	100.00	100.00	100.00	100.00
SIFT 720 aug	8.17	53.37	88.94	99.52	100.00	100.00	100.00
SIFT 2160 aug	18.42	66.84	88.42	98.42	100.00	100.00	100.00

Fig. 9. Top-k accuracy using different methods.

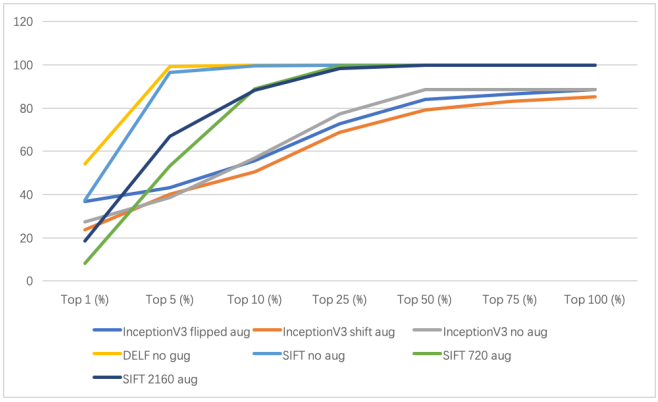


Fig. 10. Top-k accuracy using different methods.

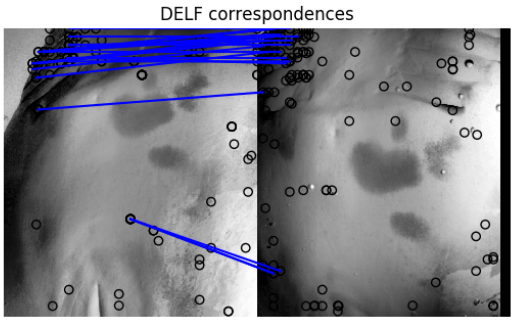


Fig. 11. A DELF matching result of manta ray Apple.

I also qualitatively compared CNN-based local feature (DELF) matching with keypoint feature matching (SIFT). Fig. 11 is a DELF matching illustration of manta ray Apple, and Fig. 12 is a SIFT matching result of Apple. From the two pictures we can observe that the two keypoint descriptors tend to focus more on gills of the rays, which have clearer edges. Apple have relatively light patterns of the apple shape, and both keypoint descriptors seem to largely neglect the "truely distinctive" pattern.

Fig. 13 and Fig. 14 are DELF and SIFT correspondence matching results of manta ray Carrot. Carrot have distinctive patterns with nice lighting and contrast condition, so both DELF and SIFT perform good matching, although we can still observe several wrong matching pairs in Fig. 14. DELF descriptors here seem to ignore some pattern matches in peripheral areas, while SIFT can still catch matching features in these areas.

Fig. 15 shows a good matching result using InceptionV3 model. We can see Alien has dark and distinctive patterns easy to capture, and the images have similar illumination, viewpoints, and contrast conditions. These possibly make the identification and matching results so desirable.

Fig. 16 shows a hard matching task of manta ray Gonzales, since both of its images have large occlusion by fish. The SIFT detectors captured many features of the fish, making wrong matches because of the fish's fast movement. The distinctive patterns of Gonzales also got largely occluded, making the matching task hard.

Computation times per image are shown in Fig. 17. SIFT computation is clearly faster than feature computation by neural networks (both InceptionV3 and DELF methods). Compared with previous accuracy results, we can safely reach the conclusion that SIFT does well in manta ray identification tasks, considering time-accuracy tradeoffs.

V. Conclusion and Future Work

In this study I have described and compared three techniques for automated identification of manta rays. The methods are reasonably robust to changes in some degrees of viewpoint, scale, lighting conditions, pose, and occlusions, getting pretty nice top-25 accuracy results (close to 100% for traditional SIFT and deep-learned DELF keypoint descriptors), with the majority of rays being correctly identified in the top-ranked images. Unlike many other automated matching techniques, the three approaches require only minimal user effort. This project further demonstrates that even species such as manta rays, whose characteristic markings are often indistinct and show substantial variability, can successfully be matched using automated photographic identification techniques, provided that the methods are so-

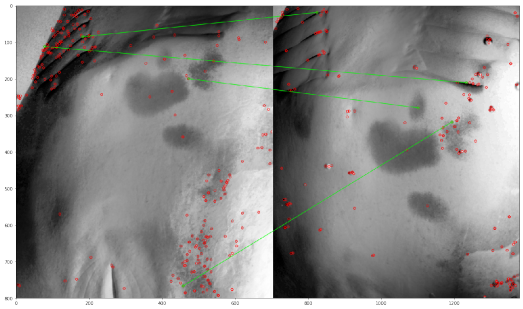


Fig. 12. A SIFT matching result of manta ray Apple.

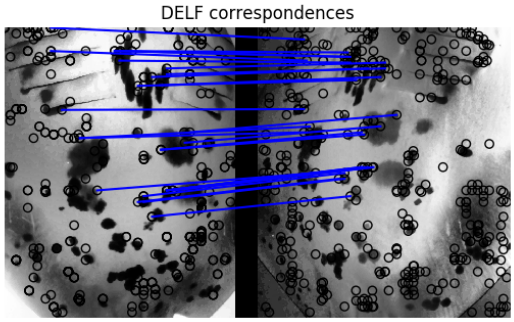


Fig. 13. A DELF matching result of manta ray Carrot.

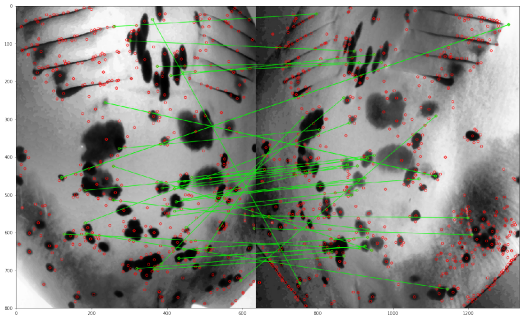


Fig. 14. A SIFT matching result of manta ray Carrot.

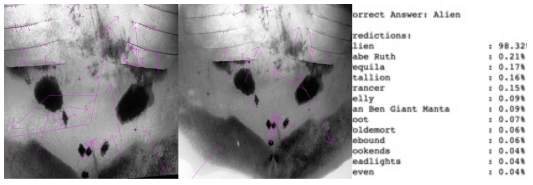


Fig. 15. Two pictures of manta ray Alien with SIFT features, and an InceptionV3 matching result of Alien.

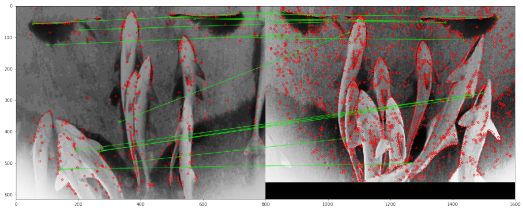


Fig. 16. Two pictures of manta ray Gonzales with SIFT matching result.

Method	SIFT	InceptionV3 (test time/batch)	DELF
Time/s	0.009	1.364	2.895

Fig. 17. Computation time using 2.6 GHz Intel Core i7 with RAM of 16 GB 2400 MHz DDR4.

phisticated enough to deal with the highly diffuse nature of the spot patterns and the challenges of underwater imagery.

The comparison allows us to draw several conclusions. First, the DELF descriptor outperforms SIFT on the matching and identification task, and the other CNN method, InceptionV3, only outperforms SIFT on the top-1 accuracy results. The experiment on different augmentation transformations may indicate a limitation of the CNN originally trained on ImageNet, and more experiments such as on batch size may be needed. Some good and bad performances of the three methods were discussed. The computational cost is in favor of SIFT.

SIFT is very interesting and promising for tasks where speed and simplicity are of major importance. However, for some computer vision tasks that rely on descriptor matching, especially top-1 matching, it is worth considering the use of features trained with convolutional neural networks.

By completing this identification task, I contributed to demonstrate that modern pattern recognition techniques are powerful tools for ecological research, and there are other ways of achieving good results apart from SIFT methods as used in current state-of-art systems. By contributing to animal identification, which helps to enhance scientists' ability to track individual animals, more fine-grained information can be provided for fisheries management, and visually assessable parameters such as anthropogenic scarring provide vital information for ecological impact assessments and action planning.

Apart from InceptionV3 and the pretrained DELF model, the influence of other base network selections and training methods can be further explored. More experiments on CNN parameters and batch sizes are desired. In addition, evaluating identification with consideration of investigating the use of ancillary identifying information such as sex, maturity, color morphism (melanism or leucism), or scars, are desired to be performed. Investigating the suitability of the three methods on other species are also needed.

ACKNOWLEDGMENT

The author gratefully acknowledges data support and insightful advice from Dr. Chris Town, as well as supervision support from Dr. Marwa Mahmoud and Prof. John Daugman.

REFERENCES

[1] Couturier, L. I. E., et al. "Biology, ecology and conservation of the Mobulidae." Journal of fish biology 80.5 (2012): 1075-1119.

[2] Marshall, A. D., and S. J. Pierce. "The use and abuse of photographic identification in sharks and rays." Journal of fish biology 80.5 (2012): 1361-1379.

[3] Stewart, Joshua D., et al. "Research priorities to support effective manta and devil ray conservation." Frontiers in Marine Science 5 (2018): Article-number.

[4] Town, Christopher, Andrea Marshall, and Nutthaporn Sethasathien. "M anta M atcher: automated photographic identification of manta rays using keypoint features." Ecology and evolution 3.7 (2013): 1902-1914.

[5] Burghardt, Tilo, and Neill Campbell. "Individual animal identification using visual biometrics on deformable coat patterns." 5th International Conference on Computer Vision Systems (ICVS). 2007.

[6] Mortensen, Eric N., et al. "Pattern recognition for ecological science and environmental monitoring: An initial report." Algorithmic Approaches to the Identification Problem in Systematics (2007): 12.

[7] Moskvyak, Olga, and Frederic Maire. "Learning Geometric Equivalence between Patterns Using Embedding Neural Networks." 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA). IEEE, 2017.

[8] Moskvyak, Olga, et al. "Robust Re-identification of Manta Rays from Natural Markings by Learning Pose Invariant Embeddings." arXiv preprint arXiv:1902.10847 (2019).

[9] Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.

[10] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." European conference on computer vision. Springer, Berlin, Heidelberg, 2006.

[11] Hansen, Mark F., et al. "Towards on-farm pig face recognition using convolutional neural networks." Computers in Industry 98 (2018): 145-152.

[12] Nepovinnykh, Ekaterina, et al. "Identification of Saimaa Ringed Seal Individuals Using Transfer Learning." International Conference on Advanced Concepts for Intelligent Vision Systems. Springer, Cham, 2018.

[13] Bogucki, Robert, et al. "Applying deep learning to right whale photo identification." Conservation Biology (2018).

[14] Donahue, Jeff, et al. "Decaf: A deep convolutional activation feature for generic visual recognition." International conference on machine learning. 2014.

[15] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[16] Moya, scar, et al. "APHIS: a new software for photo-matching in ecological studies." Ecological Informatics 27 (2015): 64-70.

[17] Daz-Calafat, Joan, et al. "Individual unique colour patterns of the pronotum of Rhynchophorus ferrugineus (Coleoptera: Curculionidae) allow for photographic identification methods (PIM)." Journal of Asia-Pacific Entomology 21.2 (2018): 519-526.

[18] Ranguelova, Elena, Mark Huiskes, and Eric J. Pauwels. "Towards computer-assisted photo-identification of humpback whales." 2004 International Conference on Image Processing, 2004. ICIP'04.. Vol. 3. IEEE, 2004.

[19] Gilman, Andrew, et al. "Computer-assisted recognition of dolphin individuals using dorsal fin pigmentations." 2016 International Conference on Image and Vision Computing New Zealand (IVCNZ). IEEE, 2016.

[20] Bolger, Douglas T., et al. "A computerassisted system for photographic markrecapture analysis." Methods in Ecology and Evolution 3.5 (2012): 813-822.

[21] Parkhi, Omkar M., Andrea Vedaldi, and Andrew Zisserman. "Deep face recognition." bmvc. Vol. 1. No. 3. 2015.

[22] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[23] Babenko, Artem, et al. "Neural codes for image retrieval." European conference on computer vision. Springer, Cham, 2014.

[24] Tolias, Giorgos, Ronan Sicre, and Herv Jgou. "Particular object retrieval with integral max-pooling of CNN activations." arXiv preprint arXiv:1511.05879 (2015).

[25] Arandjelovic, Relja, et al. "NetVLAD: CNN architecture for weakly supervised place recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[26] Verdie, Yannick, et al. "TILDE: a temporally invariant learned detector." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[27] Moo Yi, Kwang, et al. "Learning to assign orientations to feature points." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[28] Han, Xufeng, et al. "Matchnet: Unifying feature and metric learning for patch-based matching." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[29] Zagoruyko, Sergey, and Nikos Komodakis. "Learning to compare image patches via convolutional neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[30] Yi, Kwang Moo, et al. "Lift: Learned invariant feature transform." European Conference on Computer Vision. Springer, Cham, 2016.

[31] Noh, Hyeonwoo, et al. "Large-scale image retrieval with attentive deep local features." Proceedings of the IEEE International Conference on Computer Vision. 2017.

[32] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[33] Long, Jonathan L., Ning Zhang, and Trevor Darrell. "Do convnets learn correspondence?." Advances in Neural Information Processing Systems. 2014.

[34] Noh, Hyeonwoo, et al. "Large-scale image retrieval with attentive deep local features." Proceedings of the IEEE International Conference on Computer Vision. 2017.

[35] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[36] Bentley, Jon Louis. "Multidimensional binary search trees used for associative searching." Communications of the ACM 18.9 (1975): 509-517.

[37] Jegou, Herve, Matthijs Douze, and Cordelia Schmid. "Product quantization for nearest neighbor search." IEEE transactions on pattern analysis and machine intelligence 33.1 (2011): 117-128.

[38] Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." Communications of the ACM 24.6 (1981): 381-395.

[39] https://opencv.org/

[40] Marshall, A. D., C. L. Dudgeon, and M. B. Bennett. "Size and structure of a photographically identified population of manta rays Manta alfredi in southern Mozambique." Marine Biology 158.5 (2011): 1111-1124.

[41] https://github.com/tensorflow/models