Visual fingerprinting for lobsters

Parmeet Singh
Dalhousie University
Halifax,Nova Scotia
Email: pr819318@dal.ca

Dr.Mae Seto
Dept. of Mechanical Engineering
Dalhousie University
Halifax,Nova Scotia
Email: mae.seto@dal.ca

***Abstract*—Recognition of individual marine animals is often required in experiments to monitor their growth and experiment. The problem is usually solved by tagging individual with uniquely coded tags. However, the process of tagging can cause physical harm or restriction in the movement of a lobster and eventually defeat the purpose of the tagging. Lobsters periodically shed their exoskeletons which makes the tagging process all the more difficult. This paper aims to identify individual lobsters based upon the pattern on exoskeleton of the lobster. We use different convolutional neural network architectures towards that end.**

***Keywords*-Convolution neural networks; One-shot learning; lobsters; pattern-matching;**

I. Background

Deep learning has shown remarkable results in the human face recognition. Several social network platform can tag photos automatically without human intervention. The paper explores the face recognition techniques to lobster recognition.

A regular neural network consists of a series of hidden layers. The input to a network is a single vector which is sequentially modified by the hidden layers. Each hidden layer consists of neurons that provide an output value from applying a function to the input values from the receptive field in the prior layer. This function is in the form of a vector of weights and a bias. The hidden layers are also fully-connected layers because each neuron in the one layer is connected to every other neuron in the next layer. Learning is achieved through incremental changes to these weights and bias via back propagation. Convolutional neural networks(CNNs) are a category of neural networks that modify input volumes with layers like convolution layers, pooling layers and fully connected layers. Convolution layers contain a set of learnable filters that slide across the width and height of the input volume during the forward pass. Pooling layers downsample the spatial dimension of the input volume through averaging or max pooling.

Acquiring enough labelled data for image classification using supervised CNN can be a challenge. This can be mitigated by re-using models trained on different image sets [16]. VGG16: These are deep convolutional networks trained by the Visual Geometry Group proposed by Simonyan and Zisserman [10]. Their network uses 3×3 convolutional layers stacked on top of each other. The first step is a convolution of the image. Then, the image size is reduced through down sampling (max pooling). This alternates until the two layers become fully connected.

II. Related work

MacDiarmid et al[8] performed a study on identification of individual spiny lobsters. They observed that lobster body patterns are preserved at ecdysis(when lobster cases off its shell) which enables identification of individual spiny lobster. MacDiarmid et al obtained 15 male and female Jasus edwardsii lobsters and marked them with color coded tags. They obtained high quality images of lobsters before and after moulting and observed that lobster body patterns are retained with slight modification and elaboration of patterns. They also observed that the body patterns considerably vary among individuals.

Gosselin[2] et al. examined a set of 332 female crabs and observed that each crab retained the pattern on the carapace after moulting(when the crab casts of the shell). They employed three inexperienced observers to match the exuviae(the cast shell of the lobsters) to photos of crabs based upon the pattern on their carapace indicating that it is possible to identify individual lobsters based upon the carapace pattern.

Hillman et al.[4] developed a method to identify individial marine animals by comparing new images with a group of previously identified images in the database. The matching algorithm was based upon the pattern of nicks and notches on the dorsal fins of the marine animals. They extracted the boundary of the fin and used curve matching to compare the shape of the fin from the existing database.

One of our datasets has approximately five images per lobster. However,deep learning approaches require large datasets to perfom. One shot or few shot classification techniques are useful when there is less data for the classifier to train upon. Koch [7] et al use siamese networks for one shot classification of images. A siamese network consists of a two convolutional neural networks. The two CNNs have shared weights. Pairs of images of same class and different classes are created in equal proportion in a single batch of training. The pair of images is fed into the CNNs in the siamese network. The structure of the CNN is a series of

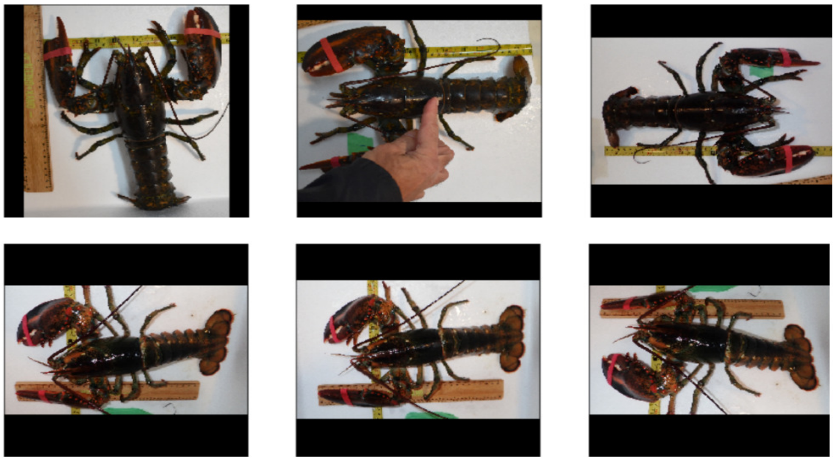


Figure 1. Variation in carapace patterns on lobster: The top row contains images of the same lobster in different views. Similarly, the bottom row contains images of same lobster in different views. The top and bottom lobster can be visually discriminated by the pattern on the carapace and claw.

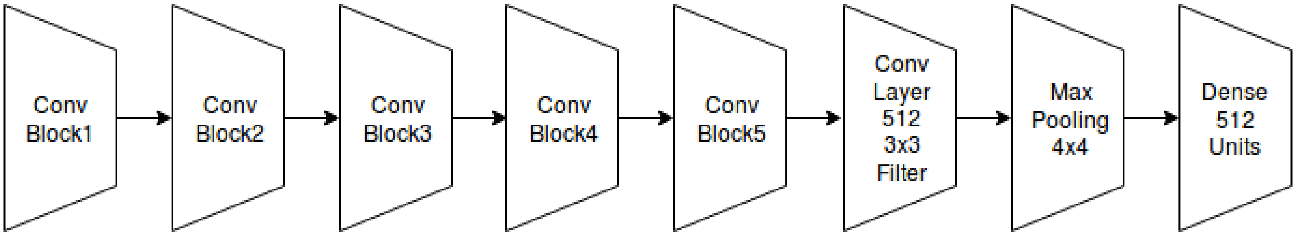


Figure 2. Modified VGG16 network: The VGG-16 network is modified by removing the two fully connected layers at the end and adding two 512 3×3 convolution layers and max-pooling layers. Sunsequently, dense layer containing 512 hidden units were added.

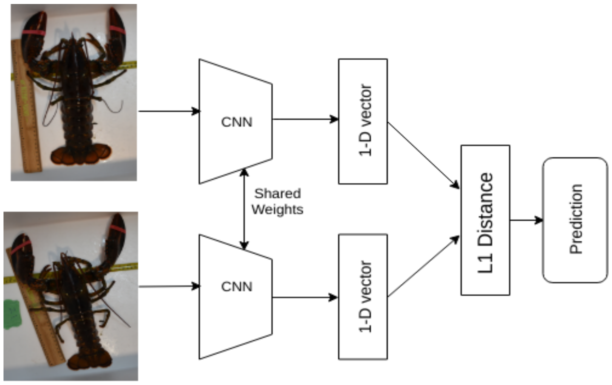


Figure 3. Siamese Networks: Pairs of images are fed into the siamese network. The Siamese network has two convolution neural networks which are fed two separate images. The siamese learns to differentiate between pairs of same classes and different classes.

convolution and pooling layers. The feature representations from the layers are flattened to a one-dimensional vector. The distance between the two vectors is calculated using the L1 or L2 distance. The calculated distance is fed into a fully connected layer and then finally optimized using the cross entropy loss function. The output label is 1 is the images are from the same class and 0 otherwise.

Taigman et al. [15] developed a nine-layer convolution neural network that is trained on a dataset of four million images having approximately 4000 face identities. They also developed an face alignment system based upon a 3-dimensional model of faces.

Sun et al.[13] learn a high-level feature representation of faces called DeepID using deep learning for face verification. DeepId features were extracted from the last hidden layer activations of the CNNs. Sun et al. used 60 CNNs that were trained on different patches around the five landmarks of faces i.e two eyes,nose and two mouth corners. The CNNs were trained on a set of 10,000 images. Sun et al. claimed that any state-of-the-art classifiers can be learned based upon these representations.

Sun et al.[11] developed a model DeepID2 to reduce intra-class variations and increase inter class variations using both face-identification and face verification as supervision. The input data consisted of pairs of images which was fed to a two convolution neural networks with shared weights. DeepID2 is optmized using two loss functions. The first loss function is for identification which classifies output of each CNN to n different identities where n is the number of faces the CNN classifies. The identification loss CE 1 is the cross-entropy loss given by:

$$CE = -1 * \sum_{i=1}^{n} c_i * log(\hat{c}_i) \tag{1}$$

where c_i is zero for all i except the target class t for which it is 1. $\hat{c}_i$ is the target probability distribution. The second loss function is the verification loss which discriminates between the feature representations of the two input images. The verification loss is a measure of contrast between the images. The contrastive loss was given by [3] is defined as :

$$Loss = \begin{cases} \frac{1}{2}||f_i - f_j||_2{}^2 & \text{if } y_{ij} = 1 \\ \frac{1}{2} * max(0, m - ||f_i - f_j||_2{}^2) & if y_{ij} = 0 \end{cases} \tag{2}$$

Here, f_i and f_j are representations from image i and image j. $y_{ij} = 1$ when both images are of same category and $y_{ij} = 0$ when both images are of different category. The equation 2 requires the distance larger than a margin m.

Sun et al [12] developed DeepID3 by modifying two deep neural network architectures VGG[10] and Inception[14] for face recogntion. Similar to [11], they used face identification Eq. 1 and face verification Eq. 2 supervisory loss functions during training. Zhou et al. [17] trained a 10-layer CNN on a much larger dataset of faces than DeepId [11] without joint verification and identification as well no 3-D alignment. They achieved 99.50 percent accuracy of LFW[6] dataset(start of the art during that time) using the aforementioned dataset that had five million labeled faces with about 20,000 individuals. Schroff et al. [9] presented a CNN architecture called FaceNet in which triplet loss was used. The siamese network was modified to take as input three images. The triplet images contained an anchor image,positive image and a negative image. The anchor image and positive image have the same identity whereas negative is different from both the anchor and positive image. The triplet loss aims to minimize the distance between the anchor and the positive image and maximize the difference between negative and anchor image. However, it is tricky to choose positive and negative pairs that are closely different from each other. Naive sampling can lead choosing easy positive and negative pairs in the batch leading to collapse in training. Many of the aforementioned CNN architectures such as DeepID3 have been trained upon large amounts of data. We try out siamese network based architecture along with pretrained VGG2 as building blocks for the lobster recognition problem.

III. Methodology

A. Dataset

The dataset contains 560 images of 114 lobsters which is approximately 5 images per lobster. This dataset was augmented using the *imgaug* [1] software image augmentation library. The following image augmentations were performed: randomly rotated horizontally and vertically; affine transformations like image translation from -10% to 10%; rotations from -45° to 45 °.

As part of the data preparation for the training, the images were re-sized to 448×448 pixels for input into the CNN.

B. Model Architectures

1) Softmax-based classifier: A modified VGG-16 network Fig.2 is used where the two fully connected layers are removed and two convolution layers with 512 3×3 filters and two max-pooling layers are added. Subsequently, a dense layer containing 512 units is added. The final layer is dense layer with units equal to the number of class ids and softmax activation. The softmax function changes an n-dimensional vector z of arbitrary real values to an n-dimensional vector $\sigma(z)$ where $\sum_{i=1}^{K} \sigma_i = 1$ and $\sigma(z_j) = \frac{e^{(}z_j)}{\sum_{k=1}^{K} e^{(}z_k)}$.

2) Siamese Network: A modified VGG-16 network2 is used where the two fully connected layers are removed and two convolution layers with 512 3×3 filters and two max-pooling layers are added. Subsequently, a dense layer containing 512 units is added.

In each batch we create pairs of images such that half of the pairs are of the same lobster(but from a different view or camera) and the other half are images of different lobsters. Each image in the pair is used an input to the aforementioned CNN. 3. The CNN network gives a 512 dimensional representation of the each image. The euclidean distance between the two vectors is calculated. The objective of the siamese network is to minimize the euclidean distance if the image pairs is of same class and maximize the euclidean distance if they are of different class.

We experimented with two loss functions for optimizing the distance between image pairs which is the binary cross-entropy function1 and contrastive loss function2.

C. Inference

A test image is classified by calculating the distance of the test image from each of the lobsters in the training set. The class of image in the training set to which the test image is shortest in distance to is the class of the test image.

IV. Results

The models are trained for 200 epochs with a patience number of 50 i.e if the validation loss does not decrease for a run of 50 epochs the training process is terminated. The dataset split such that training was carried on 75 percent data and test accuracy was reported on the other 25 percent.

Table I reports the top-k accuracy of softmax-based classifier and metric based classification i.e cross-entropy and contrastive loss. The top-k accuracy is the percentage of instances where true predicted labels lies in the top-k probabable predictions for that instance. The top-1 accuracy

```
Data:
M(x) = embedding for image x
$i$ = test image
Result: Find the class id of the test image
$support\_set \leftarrow unique\ labels\ in\ the\ training\ set$
$em_i \leftarrow M(i)$
for $k\ in\ support\_set$ do
    $sum_k \leftarrow 0$
    $count_k \leftarrow 0$
    for $image\ j\ with\ label\ as\ k$ do
        $em_j \leftarrow M(j)$
        $d_{ij} \leftarrow || em_i - em_j ||$
        $sum_k \leftarrow sum_k + d_{ij}$
        $count_k \leftarrow count_k + 1$
    end
    $average_k \leftarrow sum_k / count_k$
end
$predicted\_label \leftarrow argmin(average_k)$
```

Algorithm 1: Algorithm to find the closest label to the given test image

of cross-entropy and contrastive loss based methods are marginally better than the softmax-based classifier. The top-3 and top-5 accuracy of contrastive loss based methods are better than the softmax-based classifier and cross-entropy based classifier.

Table I
COMPARISON OF TOP-K ACCURACIES USING VARIOUS LOSSES FOR ONE-SHOT CLASSIFICATION. THE CONTRASTIVE LOSS BASED CLASSIFIER HAS BETTER TOP-3 AND TOP-5 ACCURACY COMPARED TO SOFTMAX-BASED AND CROSS-ENTROPY CLASSIFIER.

Model	Top-k accuracy				
	k=1	k=3	k=5	k=7	k=9
Softmax-based classifier	0.54	0.70	0.79	0.81	0.83
Cross-entropy	0.56	0.76	0.85	0.91	0.96
Contrastive Loss	0.59	0.84	0.90	0.91	0.94

Figure 4 shows the variation in test accuracy with variation in the margin parameter of the contrastive loss function. The top-k accuracies increase with increasing margin parameter upto a certain point. Setting a very low value of margin makes the loss function not care about different pairs of images i.e second part in eq.2. However, setting a high value of margin gives alot of weightage to the not same pairs and not care about same pairs.

Figure 5 shows the euclidean space of features after applying principal component analysis of the feature space of the dataset embeddings learned by the various classifiers. Principal componant analysis reduces the dimensionality of a dataset that contain large number of features to a dataset that contains the most of the information in the large dataset. The metric based features i.e cross-entropy and constrastive loss have more inter-clusters distance and lesser intra-cluster distances compared to softmax based features.

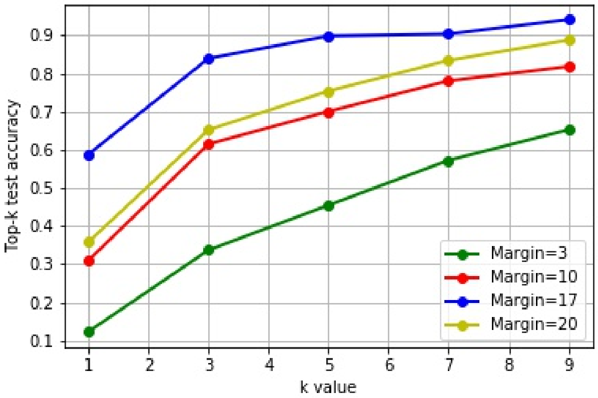


Figure 4. Variation in test accuracy upon varying the margin hyperparameter. Increasing the margin value increases the test accuracy upto a point until it starts decreasing after a certain point.

V. Conclusion

The paper demonstrates the use of convolutional neural networks to identify individual lobsters over a set of lobsters. Horiguchi et al. [5] state that metric based features perform well when there is less data i.e one-shot or few shot classification. This is further validated using our result. Since we had around five images per lobsters(less data), the metric based features perform better than softmax based features.

References

[1] Aleju. aleju/imgaug, July 2015.

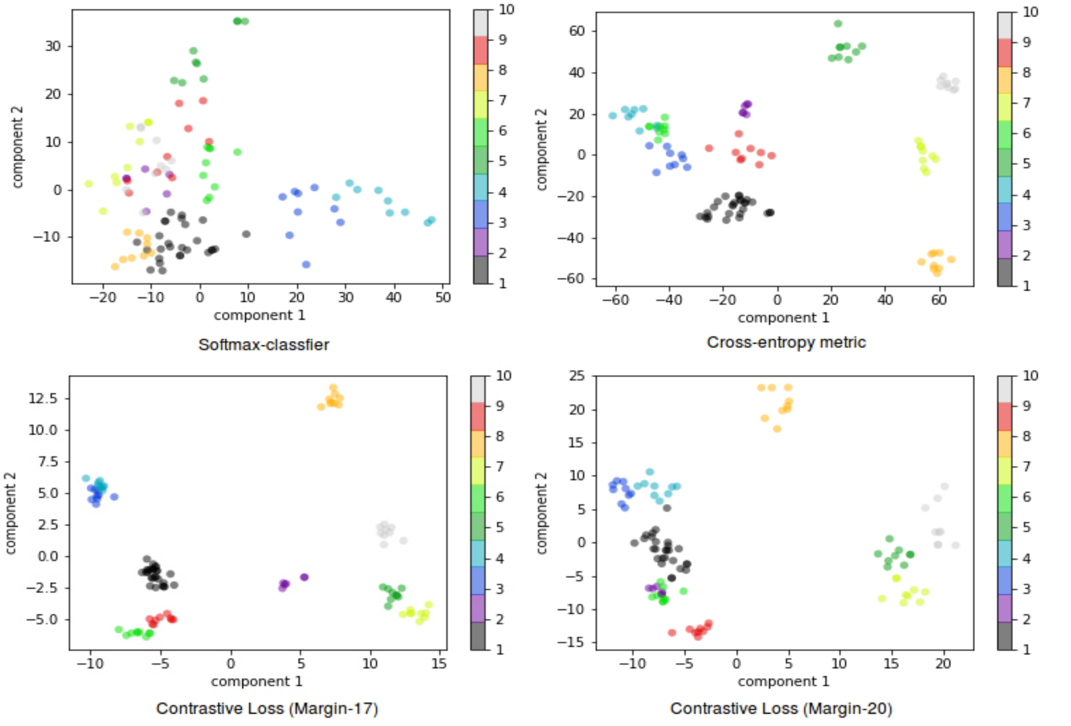
[2] Thierry Gosselin, Bernard Sainte-Marie, and Jean-Marie Sévigny. Individual identification of decapod crustaceans ii: Natural and genetic markers in snow crab (chionoecetes opilio). *Journal of Crustacean Biology*, 27(3):399–403, 2007.

[3] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *null*, pages 1735–1742. IEEE, 2006.

[4] GR Hillman, B Wursig, GA Gailey, N Kehtarnavaz, A Drobyshevsky, BN Araabi, HD Tagare, and DW Weller. Computer-assisted photo-identification of individual marine vertebrates: a multi-species system. *Aquatic Mammals*, 29(1):117–123, 2003.

[5] Shota Horiguchi, Daiki Ikami, and Kiyoharu Aizawa. Significance of softmax-based features in comparison to distance metric learning-based features. *arXiv preprint arXiv:1712.10151*, 2017.

[6] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.



Softmax-classfier

Cross-entropy metric

Contrastive Loss (Margin-17)

Contrastive Loss (Margin-20)

Figure 5. Top left image depicts the PCA analysis of the euclidean space of lobsters from the softmax classifier. The inter-class differences are less and intra-class differences are large. Top-right image:The PCA anaysis using cross-entropy loss metric has lower intra-class different and higher inter class difference and spacings. Bottom-Left: we use contrastive loss with margin 17(tuned hyper-parameter). Smaller intra-classes differences and larger inter-class difference compared to cross-entropy loss.

[7] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[8] Alison B MacDiarmid, Megan D Oliver, Robert A Stewart, and Dharini Gopal. Conservation of unique patterns of body markings at ecdysis enables identification of individual spiny lobster, jasus edwardsii. 2005.

[9] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[11] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014.

[12] Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

[13] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1891–1898, 2014.

[14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[15] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[16] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[17] Erjin Zhou, Zhimin Cao, and Qi Yin. Naive-deep face recognition: Touching the limit of lfw benchmark or not? *arXiv preprint arXiv:1501.04690*, 2015.