Underwater Fish Recognition

Name: Tsao-Wei (Brad) Huang

SUNetID: brad0309

Abstract: Computer vision algorithms can aid efforts in marine ecosystem monitoring by automating the process of recording species appearances. There are few literatures on fish recognition, and most use classical recognition methods such as PCA and LDA. This study evaluated a number of such methods, and also considered how algorithms converting color images to gray scale can affect recognition accuracy. The results showed that luminance, luster in HLV and value in HSV are the better color-to-gray algorithms for fish recognition. The results also validated previous hypothesis that using SIFT to perform object matching is a useful method for fish recognition, and its accuracy is on par with the previously proposed best algorithm using sparse representation.

1. Introduction

Long-term ecological monitoring requires a lot of labor-intensive efforts. Monitoring marine ecosystem - in particular - often involves scuba diving and collecting videos and materials manually. Recently, automatic underwater monitoring systems have been developed to observe the environment and record videos. This leads to another problem: the amount of raw data acquired is high, but probably most are sparse on information. As a result, computational approaches to extracting information from large datasets have become necessary.

Computer vision plays an important role in this process. For analysis of underwater videos, accurately capturing marine species and identifying them are the main challenges for computer vision. In this project, I will be implementing and experimenting methods in Hsiao *et al.* (2014) to understand the efficiency and accuracy of different algorithms when applied to fish detection and recognition.

2. Literature Review

2.1 Fish Recognition

There are very few literatures focusing on fish recognition, and most of them focused on using dimensionality reduction methods to extract image features and perform classification. Matai et al. (2012) evaluated how Principal Component Analysis (PCA) and Scale invariant feature transform (SIFT) can be applied to fish recognition problems.

The authors admitted that their dataset was small and did not report details of their results, but rather only expressed that SIFT, with its scale invariance, should perform better when having more training data. In addition, they hypothesized that the lower recognition rate of the PCA algorithm was due to the number of non-zero weighting coefficients of the original training images.

Hsiao *et al.* (2014) considered building a scalable pipeline that starts from deploying underwater cameras to recognizing and identifying fish species. They proposed using sparse representation classification methods to avoid the problem of having multiple training images as likely candidates with high weighting coefficients. They introduced the method of maximum probability, which considers weighting coefficients of training images belonging to a single class as estimates of the probability that the query belongs to the class. They showed that this method improved prediction accuracy when the fish images were subject to a lot of noises, and is especially useful for recognizing fishes in underwater images because of the variations in illumination, murkiness of seawater, and translucency of different wavelength lights at variable depth.

However, the above studies neglect the fact that most recognition algorithms operates on gray scale images, while colors of the fish can be important in distinguishing visually similar species. Kanan and Cottrell (2012) evaluated a number of color-to-gray scale algorithms and assessed the difference they make in face and texture recognition. They concluded that using gamma-corrected intensity (with $\gamma = 2.2$) to convert RGB images to gray scale resulted in overall best performance in face and object recognition.

2.2 Contribution of Study

In this project, I evaluated the different choices in a fish recognition pipeline, including color-to-gray algorithms, feature extraction, and classification methods (Figure 1, details see section 3). Since no previous study considered color-to-gray algorithm to be an important factor in fish recognition, I focused on this gap of knowledge and attempted to understand why certain algorithms may be better than others in fish recognition.

3. Technical Solution

3.1 Summary

As shown in Figure 1, I implemented color-to-gray algorithms, feature extraction methods and classification algorithms.

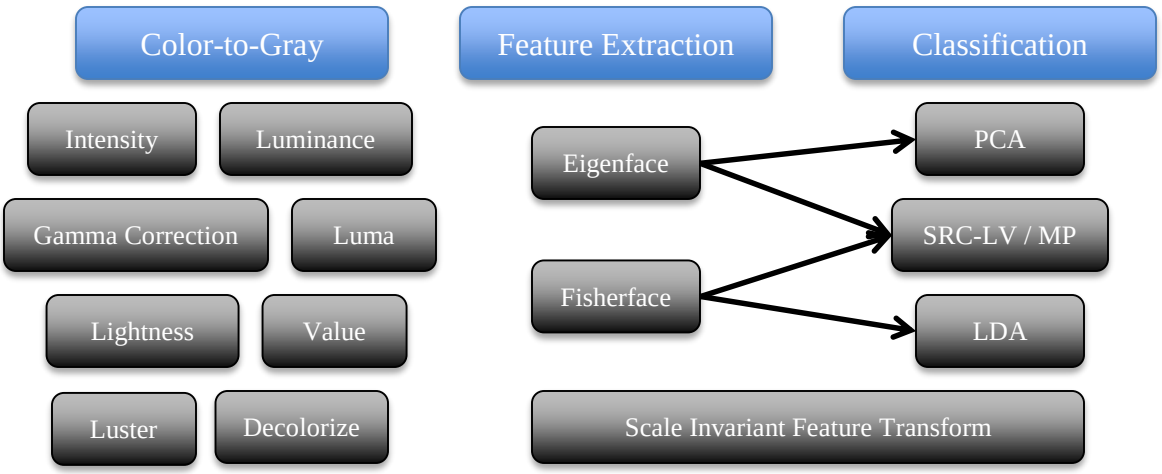

Figure 1. Algorithm Choices in the problem of fish detection.

3.2 Details

3.2.1 Dataset

The dataset was obtained from the authors of Hsiao *et al.* (2014) and consisted of 25 fish species, each having 40 images (Figure 2). I received the background subtracted and size normalized dataset (i.e. all the images are 130x180). The detection algorithm that they obtained these images was described in Hsiao *et al.* (2014), and in short is a combination of background subtraction and bounding box.

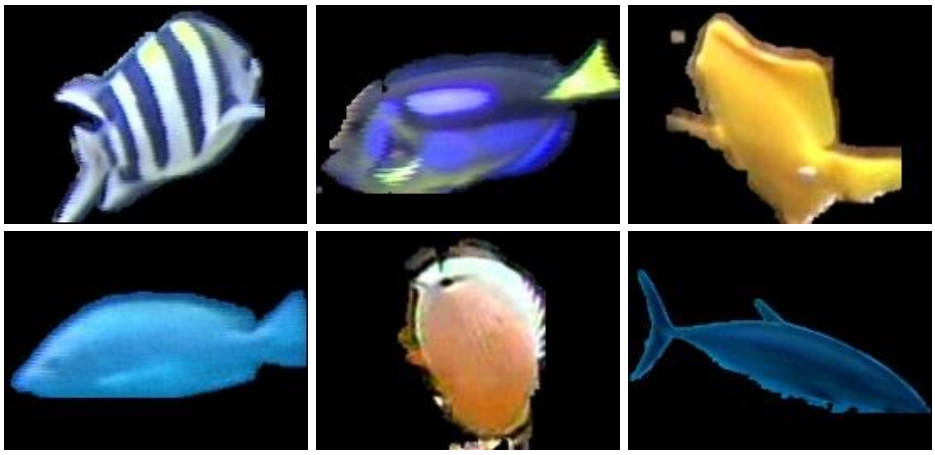

Figure 2. Sample Images from the dataset. Each image is of size 130x180 pixels in RGB format.

3.2.2 Evaluation

There are two tasks a fish recognition pipeline should do: recognition determines which species of fish the query image most likely belongs to, while identification determines whether the species of the query image is in the training data. Identification is usually implemented as rejecting a prediction when certain thresholds are not reached (e.g. distance in feature space larger than certain value). However, because it usually

Name	Equation / Process	Notes

requires a held-out set to set these thresholds and because of the time constraint, I decided to not evaluate identification rate of these methods.

The dataset is split into a train set, consisting of 20 random species each having 32 images, and a recognition test set, consisting of the remaining 8 images in the same 20 species.

3.2.3 Color-to-Gray

The color-to-gray algorithms are described in Kanan and Cottrell (2012), while the *decolorize* algorithm is described in Grundland and Dodgson (2007).

Table 1. Summary of the equations used in color-to-gray algorithms.

Intensity	$\frac{1}{3}(R+G+B)$	
Gamma-correction	$\frac{1}{3}(R'+G'+B'),$ where $t' = t^{1/2.2}$	
Luminance	$0.3R + 0.59G + 0.11B$	MATLAB rgb2gray
Luma	$0.2126R' + 0.7152G' + 0.0722B'$	High-definition TV
Lightness (CIELAB)	$\frac{1}{100}(166\, f(Y) - 16)$ $f(t) = \begin{cases} t^{1/3} & if\ t > (6/29)^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^3 t + \frac{4}{29} \end{cases}$	Perceptually uniform
Value (HSV)	$max(R,G,B)$	Absolute brightness
Luster (HLV)	$\frac{1}{2}[max(R,G,B) + min(R,G,B)]$	
Decolorize	Pixel-wise algorithm • Convert to YPQ color space • Compute gradient of PQ at the pixel • Piecewise linear mapping from the gradients and the overall saturation to gray scale value	Grundland and Dodgson (2007): Preserve and enhance color contrast, preferred by human subjects for natural images

3.2.4 Feature Extraction

Following Hsiao et al. (2014), I implemented feature extraction using Eigenfaces and Fisherfaces. Let A be the concatenation of all image pixels normalized to zero mean (so A is a 23400 x N matrix, where N = 640 is the number of training images), the Eigenfaces are the N eigenvectors with the largest eigenvalues of $A \cdot A^T$. To reduce computation time, the Eigenfaces can also be obtained by $A \cdot x$ where x are the eigenvectors of $A^T \cdot A$ (Turk and Pentland, 1991).

Fisherface is implemented similarly (Belhumeur et al. 1997). Fisherface is developed to keep variations that have discriminating power between classes and discard within class variations. First, the N - c, where c = 20 is the number of species in the training set, eigenvectors of the PCA results as described above are denoted as w_{pca}. Then, the within-class and between-class scatter matrix is then computed:

$$S_W = \sum_{j=1}^{c} \sum_{i=1}^{n_j} (x_{ij} - \mu_j)(x_{ij} - \mu_j)^T$$

$$S_B = \sum_{j=1}^{c} (\mu_j - \mu)(x_{ij} - \mu_j)^T$$

where n_j is the number of training images in the j^{th} class, x_{ij} is the i^{th} image in the j^{th} class, μ_j is the mean of the images in the j^{th} class, and μ is the mean of all training images. The generalized eigenvectors of $w_{pca}^T S_B w_{pca}$ and $w_{pca}^T S_W w_{pca}$ would be w_{lda}, and first c columns of $w_{pca} \cdot w_{lda}$ would be the Fisherfaces.

3.2.5 Classification

Several classification methods are considered: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Sparse Representation Classification (SRC) with Largest Value (LV) or Maximum Probability (MP), and a voting scheme using the Scale Invariant Feature Transform (SIFT) features.

- PCA and LDA

The PCA or LDA features of a query image are the weighting coefficients by which the Eigenfaces or Fisherfaces are linearly superimposed into the query image. This can be directly calculated as $F^T \cdot (y - \mu)$, where F is the horizontal concatenation of all the Eigenfaces or Fisherfaces. After the features of the query image are extracted, a linear search is performed to find the training image closest to the query in feature space, and the species of that training image would be assigned as the prediction (Turk and Pentland, 1991; Belhumeur *et al.*, 1997).

- SRC

SRC uses the same methods as in PCA and LDA to extract query image features. However, instead of a nearest neighbor solution, the method minimize the $l1$- or $l0$-norm of weight vector x subject to $Ax = y$, where A is the concatenation of all training features, and y is the query feature vector (Donoho 2006). $l1$-minimization is done using CVX (Grant and Boyd, 2014; Grant and Boyd, 2008). SRC enforces x to be a sparse solution, containing only a few non-zero elements, and solves the problem that with many training images close to the query in feature space, taking the species with smallest distance may not be accurate as in PCA and LDA (Wright *et al.*, 2009).

After obtaining the weights of training images, there are two methods to select the final species as the prediction. The Largest Value method selects the species of the image with the highest weight, while the Maximum Probability method sums the top few weights from each species, and selects the species with highest sum (Hsiao *et al.* 2014).

3.2.6 SIFT

The SIFT features and keypoint frames are computed using the VL_FEAT library. All of the features and keypoints from the training images are aggregated together, while keeping track of the species the images belong to. When a query image arrives, its SIFT features and keypoints are extracted, and the matches are determined using the ratio of the two closest matches with a default threshold of 0.8 (Lowe, 2004). The matches are then refined using RANSAC and only the inliers that are consistent with a single homography model are retained. After producing reliable matches, the algorithm counts the numbers of matches between the query SIFT keypoints and the keypoints in all the images of each species, and the species with the most matches would be the prediction.

4. Results

4.1 Color-to-Gray Algorithms

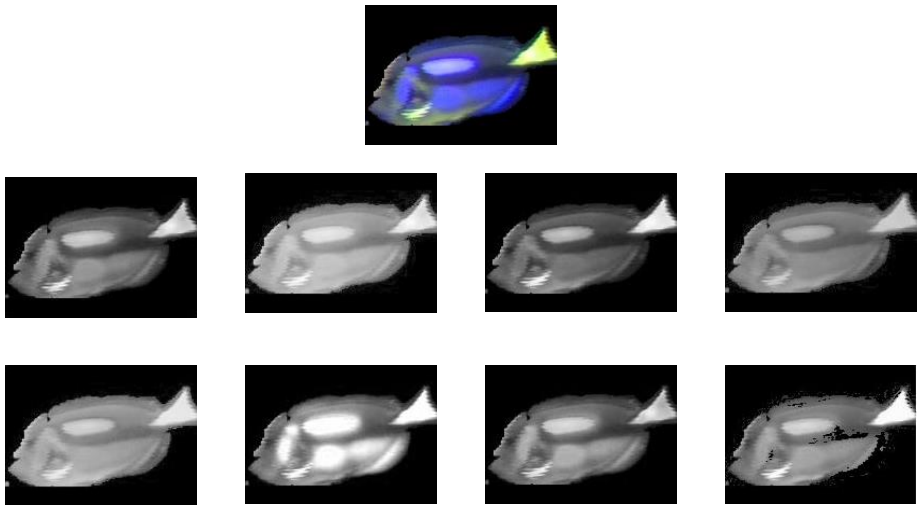

Figure 3. The color-to-gray algorithms (top to bottom, left to right): intensity, gamma-corrected intensity, luminance, luma, lightness, value of HSV, luster of HLV, and decolorize.

4.2 Feature Extraction

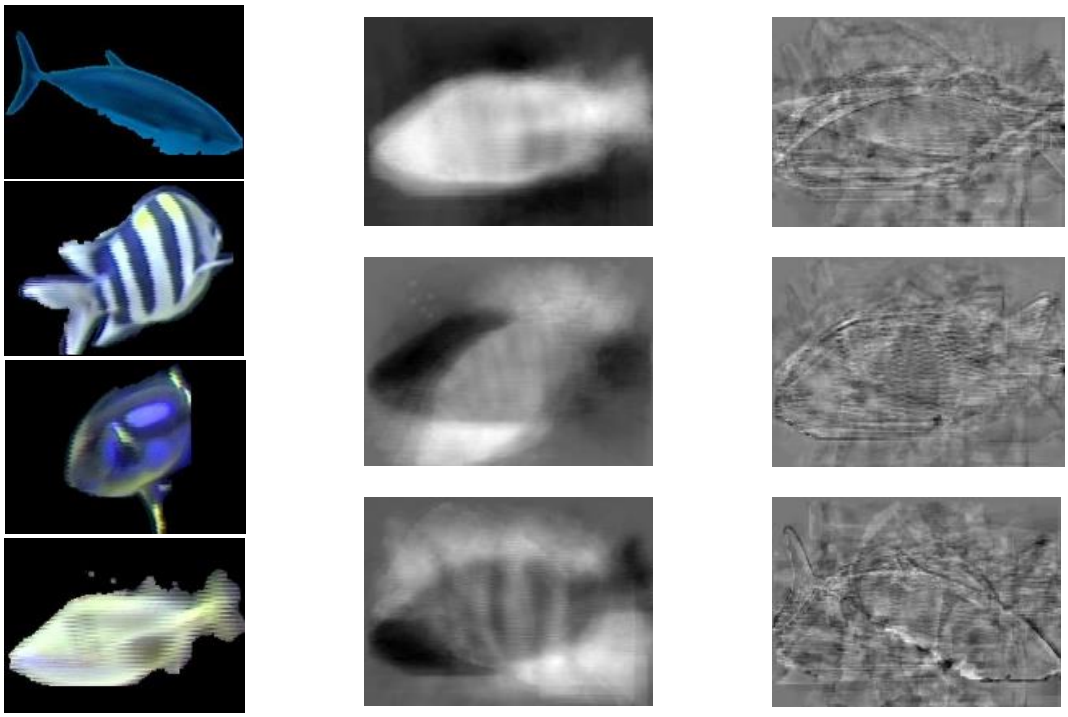

Figure 4. Several Eigenfishes and Fisherfishes obtained from training on the images of the first four species. Note that the Eigenfishes retain the basic shapes of a fish, while the Fisherfishes on between species, so the fish shape

4.3 Method Comparison: Accuracy

For every method, the recognition accuracy increases as dimensionality of the feature vector increases, but plateaus after the dimensionality reaches about 25 (Figure 5). In addition, both SRC methods are consistently better than the nearest neighbor method used in PCA and LDA, while using Maximum Probability usually increases the accuracy a little. On the other hand, SIFT does not have a feature dimensionality

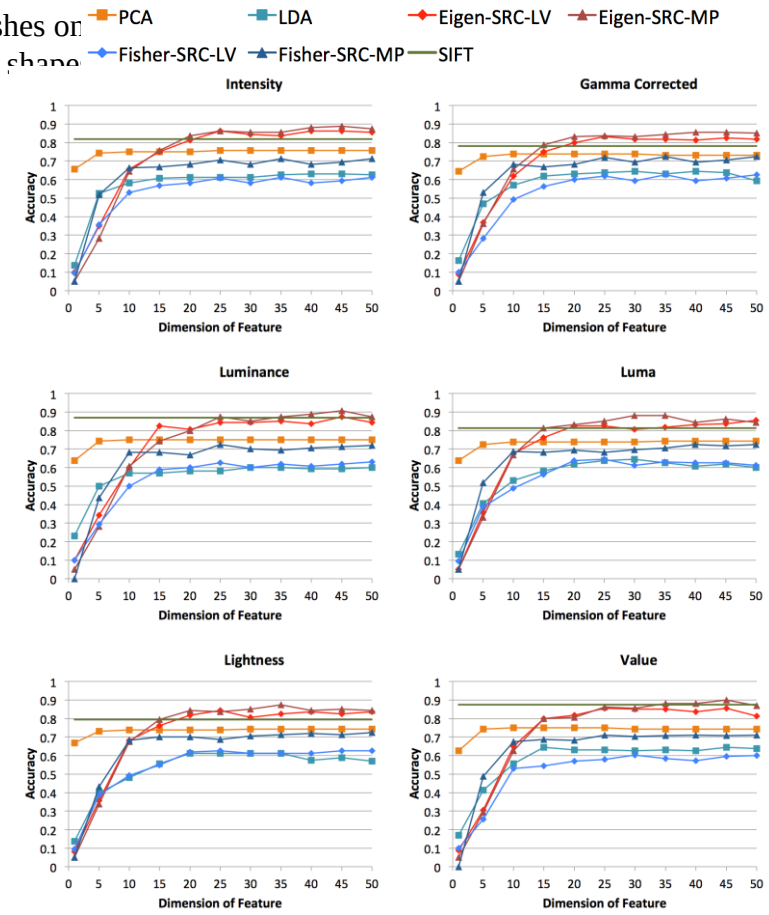

Figure 5. Accuracies of each method used on each color-to-gray algorithm over 160 test images against the dimensionality of the feature vectors.

variable, but consistently performs as well as the SRC methods.

Since the combination of Eigenfaces and SRC-MP resulted in the highest recognition accuracy, these methods are used to compute the accuracy of recognizing images produced by each color-to-gray algorithm. Among the results of Eigenfaces and SRC-MP, luminance has the best recognition accuracy (Figure 6). Kanan and Cottrell (2012) concluded that gamma-corrected intensity produced best results for face and object recognition, while luminance was best for texture recognition. These show that color-to-gray algorithms that are perceptually uniform do not make recognition easier.

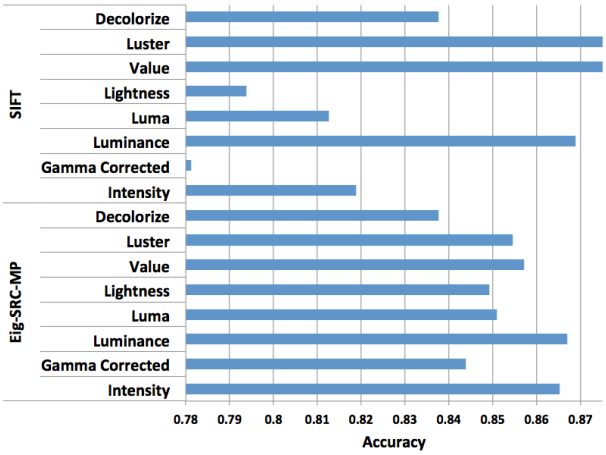

Figure 6. Classification using SIFT on gray scale images produced by luster and value results in highest accuracy, while luminance is the second highest.

In addition, SIFT, being independent of any sort of feature vector dimensionality, produces decent results (Figure 5). As a result, the highest accuracy by SIFT is being compared to other accuracies by Eigenfaces and SRC-MP. SIFT combined with using luster or value to convert RGB to gray scale results in the highest recognition accuracy. This is similar to what was hypothesized in Matai *et al.* (2012) – that the invariances of SIFT would result in the best fish recognition accuracy given a large enough dataset, because of the variations in illumination and pose of the fish images.

4.4 Method Comparison: Runtime and memory

A.

Color-to-Gray Algorithm	Runtime over 40 Images (s)
Decolorize	330.728
Others	< 1

B.

Feature Extraction	Runtime over 640 Training Images (s)

Eigenface	8.56
Fisherface	29.267
SIFT	89.489116

C.

Classification	Runtime over 160 Test Images (s)
PCA	8.56
LDA	29.267
Eigenface + SRC-LV	$62.43 + 3.67d$ $(r^2 = 0.9721)$
Eigenface + SRC-MP	$64.36 + 3.61d$ $(r^2 = 0.9718)$
Fisherface + SRC-LV	$56.49 + 3.82d$ $(r^2 = 0.9750)$
Fisherface + SRC-MP	$47.34 + 3.47d$ $(r^2 = 0.9919)$
SIFT	1074.03032

Table 2. The runtime summary of the processes. In C., $1 \leq d \leq 50$ is the dimensionality of the feature vector and the values are obtained using linear regression.

It is apparent that decolorize takes a lot longer than the other color-to-gray algorithms. The recognition accuracy using decolorized images is not high either, so decolorize is not a good solution to fish recognition problems.

For feature extraction, obtaining SIFT descriptors does take longer than Eigenfaces and Fisherfaces. On average, each image produced about 50 keypoints, and the resulting features are about 200MB for 640 species.

Lastly, classification methods vary a lot on the runtime. The PCA and LDA methods can be really fast, on the magnitude of seconds, as the only thing the methods do is a linear search over the 640 training images. On the other hand, the runtimes of the SRC methods are on the magnitude of 100 seconds, and is roughly linear against the dimensionality of the feature vectors. The runtime is mostly spent on the CVX solver, and increases as dimensionality increases since Euclidean distance in the feature space is frequently computed.

5. Conclusion

The results of these algorithms show that fish recognition is highly influenced by the algorithms converting color images to gray scale images. Depending on the classification method, different color-to-gray algorithms would perform better, and the difference can be as large as 10% recognition rate. The results also validate the use of

SIFT features on the fish recognition problem. Although the fish images are of low resolution, making SIFT using edge detectors less robust, SIFT still performs as well as the classical Sparse Representation methods.

However, the recognition accuracy is at best 92.5%, and most plateaus at around 86%. Compared to the other methods that have been developed to reach near 100% recognition on large datasets, this experiment is only using classical feature extraction and classification methods, and the dataset is not as large. Some possible ways to improve the accuracy include: using feature extraction methods that can address RGB color as opposed to just gray scale images, and using more complicated architecture to improve recognition rate.

References

Belhumeur P, Hespanda J, and Kriegman D. 1997. Eigenfaces versus Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **19**(7): 711-720.

Donoho D. 2006. For Most Large Underdetermined Systems of Linear Equations the Minimal l1-Norm Solution Is Also the Sparsest Solution. *Comm. Pure and Applied Math.*, **59**(6): 797- 829.

Grant M and Boyd S. 2008. Graph implementations for nonsmooth convex programs. *Recen Advances in Learning and Control* (a tribute to Vidyasagar M), Blondel V, Boyd S and Kimura H, eds., 95-110, "Lecture Notes in Control and Information Sciences". Springer.

Grant M and Boyd S. 2014. CVX: Matlab software for disciplined convex programming, version 2.1 beta. http://cvxr.com/cvx.

Grundland M and Dodgson NA. 2007. Decolorize: Fast, contrast enhancing, color to grayscale conversion. *Pattern Recognition*, **40**(11): 2891-2896.

He X, Yan S, Hu Y, Niyogi P, and Zhang H. 2005. Face Recognition Using Laplacianfaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **27**(3): 328-340.

Kanan C and Cottrell GW. 2012. Color-to-Grayscale: Does the Method Matter in Image Recognition? *PLoS ONE*, **7**(1) e29740.

Lowe DG. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, **60**: 91-110.

Matai K, Kastner R, Cutter GR and Demer DA. 2012. Automated techniques for detection and recognition of fishes using computer vision algorithm. *Report of the National Marine Fisheries Service Automated Image Processing Workshop*, 35-37.

Turk MA and Pentland AP. 1991. Face Recognition Using Eigenfaces. *Computer Vision and Pattern Recognition.* Proceedings CVPR '91., IEEE Computer Society Conference.

Wright J, Yang AY, Ganesh A, Sastry SS and Ma Y. 2009. Robust Face Recognition via Sparse Representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **31**(2): 210-227.