Automatic Habitat Mapping using Convolutional Neural Networks

André Diegues
Faculdade de Engenharia da Universidade do Porto
Underwater Systems and Technology Lab
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
Email: andrediegues@fe.up.pt

José Pinto
Faculdade de Engenharia da Universidade do Porto
Underwater Systems and Technology Lab
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
Email: zepinto@fe.up.pt

Pedro Ribeiro
Faculdade de Ciências da Universidade do Porto
CRACS/INESC TEC
Rua do Campo Alegre, 4169-007 Porto, Portugal
Email: pribeiro@dcc.fc.up.pt

***Abstract*—Habitat mapping is an important task to manage ecosystems. This task becomes most challenging when it comes to marine habitats as it is hard to get good images in underwater conditions and to precisely locate them. In this paper we present a novel technique for performing habitat mapping automating all phases, from data collection to classification, lowering costs and increasing efficiency throughout the process. For mapping habitats in a vast coastal region, we use visible light cameras mounted on autonomous underwater vehicles, capable of collecting and geo-locating all acquired data. The optic images are enhanced using Computer Vision techniques, to help specialists identify the habitats they contain (during training phase). In a later stage, we employ convolutional neural networks to automatically identify habitats in all imagery. Habitats are classified according to the European Nature Information System, an European classification standard for habitats.**

***Index Terms*—Convolutional Neural Networks, Computer Vision, Marine Habitat Mapping, European Nature Information System, Autonomous Underwater Vehicles**

I. INTRODUCTION

Many natural habitats and respective ecosystems all across the globe are being subject to high risks of destruction due to changing weather patterns, pollution, invasive species or over-exploitation [1]. In order to understand how human behavior impact these often fragile habitats one needs to quantify them through habitat mapping. Two consecutive habitat maps of the same region can help specialists assess the health of the region and determine a course of action to improve it.

The *Parque Natural do Litoral Norte* is a natural reserve that has recently been created in the coastal region of Esposende, in the north of Portugal. The OMARE [2] project aims to study the natural habitats as well as protect endangered species in that reserve. As the region spans about 80 km^2, typical approaches (divers that collect images of the bottom or "underwater sledges", i.e., camera recording while being dragged from the surface) are not effective as they have low location accuracy and require humans to characterize every collected image.

This article is a result of the project "MARINFO - Integrated Platform for Marine Data Acquisition and Analysis", supported by Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF).

In the context of the OMARE project, our approach is to employ multiple Light Autonomous Underwater Vehicles (LAUVs) [3] (see Figure 1) that use an advanced navigation package (DVL, INS) to collect a large amount of geo-referenced photos of the bottom. The employed LAUVs can be used from the coast all the way down to 100 meters while carrying a camera and strobe light.

In this paper we propose a novel technique for performing habitat mapping in submerged coastal areas using LAUVs and Convolutional Neural Networks (CNNs). Our approach consists in having multiple LAUVs traveling close to the bottom to acquire geo-referenced photos. Habitats in the photos are (partially) identified by marine biologists, using the European Nature Information System (EUNIS), which makes a supervised learning approach possible.

II. RELATED WORK

Looking at related work, Liu et al. [4] present a comparison of CNNs against Random Forest, Support Vector Machines and Fully Convolutional Networks (FCN) to classify 7 distinct natural land cover types, concluding that CNNs and FCNs performed better, although CNNs needed a large amount of samples to train to achieve similar performance.

Berthold et al.'s [5] work on seabed sediment classification using CNN and sidescan sonar imagery achieved an overall accuracy of 56%. Gòmez-Rìos et al. [6] also designed a solution to perform coral classification using CNNs.

Also, we were inspired to adapt the solution found by the Guirado et al.'s work [7] to classify marine habitats. They used a pre-trained CNN and fine-tuned the CNN to detect plant species with data augmentation.

Overall, Deep Learning (DL) models, specifically CNNs, are already being adopted to perform image classification in underwater environments but, to our best knowledge, there is no preceding work that uses CNNs to predict the EUNIS habitat type of underwater images to perform automatic habitat mapping according to EU standards.

III. Approach

Our approach can be described in 5 phases: 1) Image retrieval and image enhancement; 2) Photo annotation by marine biologists; 3) Data splitting into training, validation and test sets (to diminish the chance of over-fitting [8]), feeding the train and validation sets into a CNN with pre-calculated weights (transfer-learning), training the CNN (fine-tuning) and predicting the test set; 4) Validation of predictions by marine biologists; 5) Assessing the performance of the model on the test set. Phase 3 and later ones can be repeated to increase performance by adding more (annotated) samples to the training set.

Fig. 1: Employed LAUVs for image retrieval.

A. Image Enhancement

The performance of a CNN (or any machine learning model) relies on the quality of the data fed into it [9]. To assure the quality of the images and their correct annotation by humans, we need to address the following problems:

1) The presence of a dominant color and absence of light;
2) Lack of contrast.

To handle problem 1, we performed a color correction algorithm (white balance), designated by Max-White [10], which resulted in brighter images and more natural colors. To handle problem 2, we used the HSV color space and contrast stretching to enhance the image:

Contrast stretching is a gray level transformation technique that transforms the intensity of an image's pixels changing the contrast of a gray image. Gray pixel intensities values are contained in the interval $[0, L-1]$, where L is the number of possible gray levels of the image (typically $2^8 = 256$, as computers use 8 bits to represent a pixel intensity). We say that a gray level transformation corresponds to:

$$s = T(r) \tag{1}$$

where s is the output intensity level of applying the transformation T to the gray intensity level of the pixel r. When applying 2, we increase the dynamic range of the image resulting in higher contrast and better perception to the human eye [11].

$$s = (L-1)\frac{r - r_{min}}{r_{max} - r_{min}} \tag{2}$$

After converting RGB images to HSV, we applied 2 to the Saturation and Value channels of the HSV image, merged back these channels and converted again to the RGB color space.

B. Automatic Habitat Classification

A CNN is a supervised learning feed-forward model that contains one or more convolution, maximum pooling and fully connected layers [12], which currently is used, mostly, in image classification and computer vision problems [12]–[14].

Training a CNN from scratch requires a vast number of samples which were not available in our case. To counter this, we used a similar design of Guirado's et al. [7] CNN: using an already existing and pre-trained CNN (the VGG16 [15] CNN, in our case) together with data augmentation.

Data augmentation is used to introduce distortions in the training set and synthetically increase its number of samples, making it a powerful method when training set is relatively small, as it is in our case.

Originally, the VGG CNN was developed in the ImageNet Large-Scale Visual Recognition Challenge [15], which goal was to recognize 1000 different objects of images available in the ImageNet image database [16].

We used the weights computed for the ImageNet image database and, additionally, added Dropout [17] to the fully-connected layers to prevent the model from overfitting. We also adapted the last fully-connected layer to classify the habitats instead of the 1000 classes available on ImageNet and, finally, retrained the whole CNN.

The VGG16 CNN is composed with 16 layers: 13 convolution layers and 3 fully-connected layers which get, as input, a 224×224 RGB image and ending with 7×7 feature maps (after 5 max pooling operations) to be flattened to the first fully-connected layer. This procedure analyses about 138 million parameters [15].

To deploy the VGG16 CNN, we used the Keras and TensorFlow Python libraries. Keras serves as a high-level abstraction to TensorFlow (a machine learning framework) to build neural networks in an easy and fast way [18], [19].

The presented procedure allowed us to transfer learning from the VGG16 and update the weights to serve our cause.

IV. Tests and Results

In scope of the OMARE project, image retrieval missions were made with LAUVs: Noptilus 1, 2 and 3 (see Figure 1), in order to map the habitats of the coastal area of Esposende, after ensuring good weather conditions, including sea state and water turbidity. According to these conditions, the retrieved data quality may vary and, consequently, affect the performance of the CNN model.

The LAUVs were deployed using `Neptus`, the command and control software tool used to plan, execute and monitor the

LAUVs' missions (see Figure 2). Through `Neptus`, operators could also preview on the field the mission's outcome and analyze the quality of the data retrieved before beginning to enhance the images.

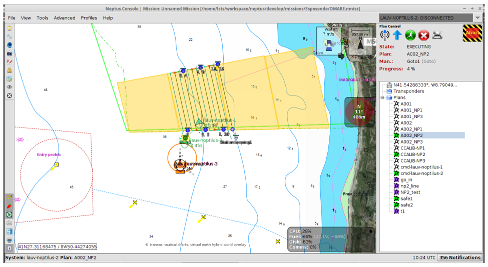


Fig. 2: `Neptus` console as used in Esposende, configured for habitat mapping applications.

After exporting the images from `Neptus`, we applied the methods described in Section III-A to enhance the images. Figure 3 contains each step of this enhancement process and the image's histogram transformations. Figure 4 provides an example of this enhancement process applied to a low-light image.

In this campaign, LAUV vehicles captured a total of 6871 images, from which marine biologists annotated 2169 images (about 32%) with, at least, an EUNIS level 3 habitat classification. The first 3 levels of an habitat classification represents the physical (energy, soil and light) properties of the habitat while the following 3 levels represent the communities and species associated to that habitat.

TABLE I: Comparison of the classification distribution and the median.

EUNIS Class	**#**
A4.1 - Atlantic and Mediterranean high energy circalittoral rock	1207
A3.1 - Atlantic and Mediterranean high energy infralittoral rock	388
A5.1 - Sublittoral coarse sediment	181
A4.7 - Features of circalittoral rock	178
A3.7 - Features of infralittoral rock	96
A5.2 - Sublittoral mud	85
A5.4 - Sublittoral mixed sediments	34
Median	**178**

Upon observing the images' classification distribution (see Table I), we have encountered an enormous discrepancy: the class A4.1 has more samples than the sum of the rest of the classifications. Also, classes A3.1 and A4.1 (related to Atlantic and Mediterranean high energy rock) differ only in the depth of the sea bottom whereas A3.1 represents the infralittoral area (with depth lower than 15 meters) and A4.1 represents the circalittoral area (with depth greater or equal than 15 meters). In the same way, A3.7 and A4.7 also share this property and since depth cannot be perceived by an image, the CNN cannot make this distinction. Therefore, we grouped classifications A3.1 with A4.1 and A3.7 with A4.7 and undersampled the data set by randomly selecting the median number of samples for each classification to balance the number of samples per classification in the dataset (see Figure 5).

After undersampling, the training set contained 654 photos (from the initial 2169 annotated images) and had 5 distinct EUNIS level 3 habitat types. The training of the CNN model was made using a Nvidia GeForce GTX 1050Ti GPU system and habitats were classified with the first 3 levels of the EUNIS. The training data was randomly split into train and validation sets (with 30% for validation) to prevent overfitting.

After 63 epochs, the model achieved a maximum validation accuracy of 92.39% in the 53^{th} epoch (see Fig. 6). The weights were saved at this stage preventing it from overfitting, given the model did not improve the validation set accuracy for the following 10 epochs.

In any ML model, the training phase (or fitting phase) can start overfitting as the model tries to represent as much as possible the data that is being trained. This process deeply downgrades the model to perform accurate predictions on new data.

Although we are already using techniques to prevent the model to overfit (train-test split, dropout, early stopping, data augmentation), DL models require a vast amount of samples in the training phase to get good results [4]. In order to evaluate the model, we sampled a separate set of 751 images chosen randomly from the data that was not labeled, asked the marine biologists to annotate this set and predicted this set.

Afterwards, we used the confusion matrix evaluation metrics (accuracy, precision, recall and F1-score) on the predictions of the test set regarding to true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) of both level 2 and level 3 of the EUNIS taxonomy:

- Accuracy: $\frac{TP+TN}{TP+FP+TN+FN}$ measures the overall success rate of the prediction;
- Precision: $\frac{TP}{TP+FP}$ measures the rating of the selected observations were correct;
- Recall: $\frac{TP}{TP+FN}$ measures the rating of the observations that should have been selected actually were selected;
- F1-score: $2 \times \frac{precision \times recall}{precision + recall}$ measures the harmonic average of the precision and recall.

When analyzing the accuracy of the model in the test set, we found out that the model is scoring lower than anticipated in comparison with the training (lower 22.8% at level 3), as we can see in Table II.

TABLE II: Overall accuracy percentage of the model.

EUNIS	**Level 2**	**Level 3**
Accuracy	85.1	69.6

As the test set contains unbalanced data (with respect to number of samples per habitat classification), we should analyze other metrics.

Tables III and IV contain the precision, recall and F1-score metrics of the predictions to the levels 2 and 3 (support corresponds to the number of samples per class). The classes (in both tables) with lower precision score than recall score are A3-A4, for level 2, and A3.1-A4.1, for level 3, as they are, coincidentally, the classes with more samples of the test set. All the remaining classes have higher precision score

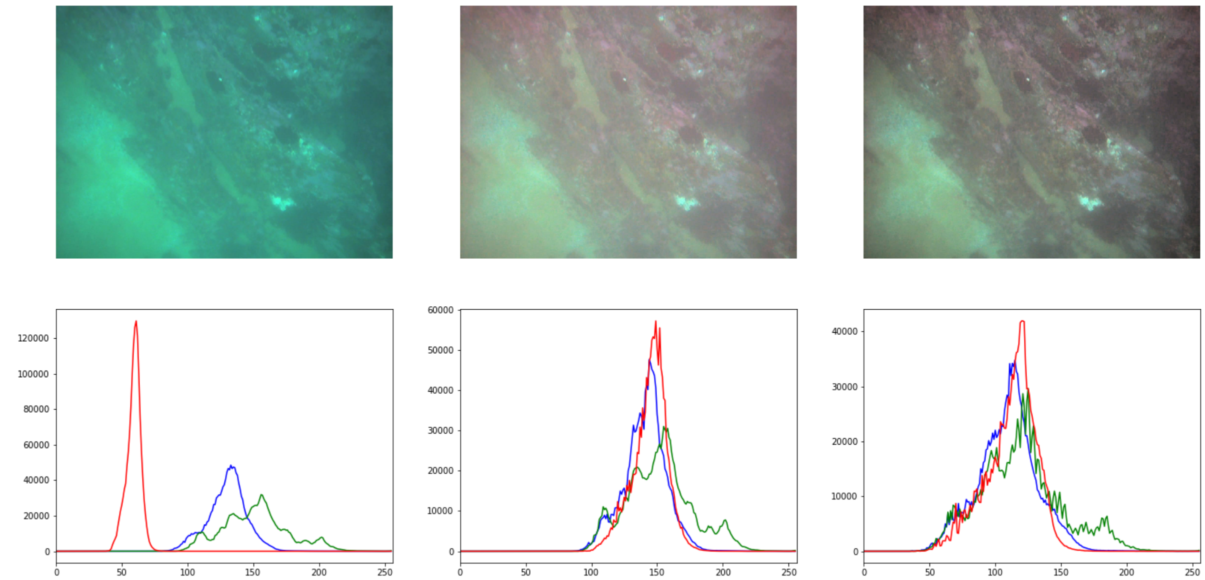
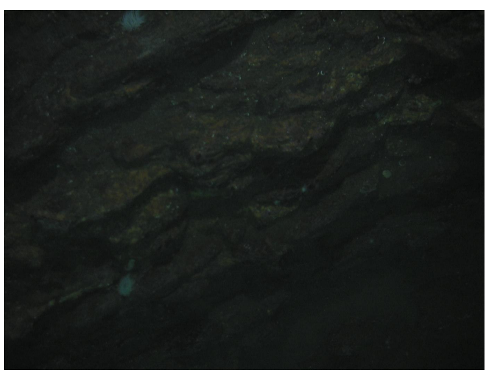
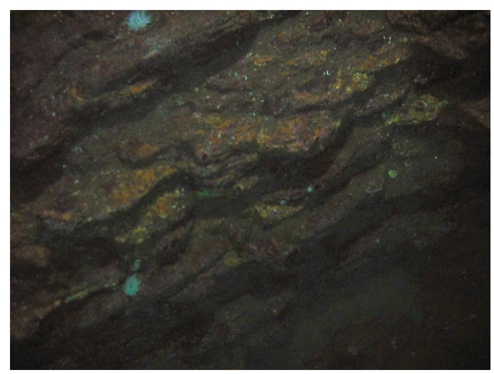


Fig. 3: Histogram analysis after transforming the original image (on the left) with the Max-White algorithm (center image) and the HSV contrast stretch (rightmost image).



(a) Retrieved image



(b) Enhanced image

Fig. 4: Max-White and HSV contrast stretching enhancement to an image retrieved by the LAUV.

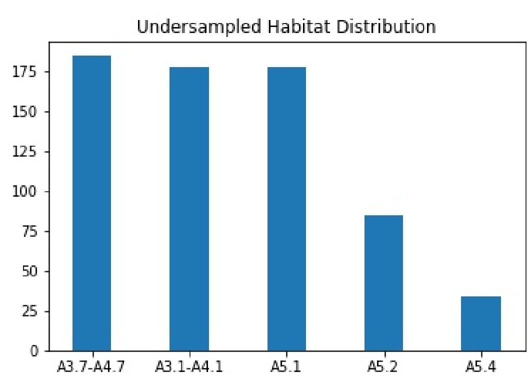


Fig. 5: Habitat classification distribution of the retrieved logs.

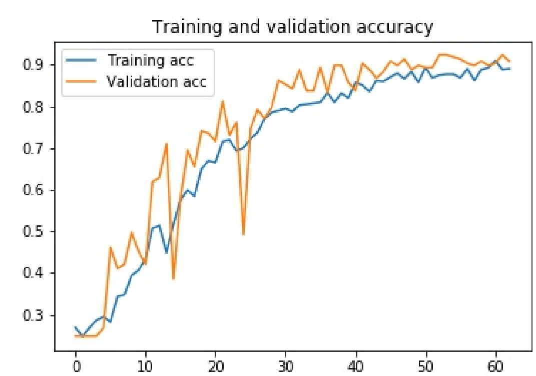


Fig. 6: Plot of the training of the CNN.

than recall score, which may indicate that classes A3-A4 and A3.1-A4.1 may have key features that are more generic and therefore the model may be overfitting for these classes.

TABLE III: Classification report on predictions of the level 2 of EUNIS.

Class	precision	recall	f1-score	support
A3-A4	0.84	0.98	0.90	536
A5	0.91	0.53	0.67	215
avg / total	0.86	0.85	0.84	751

TABLE IV: Classification report on predictions of the level 3 of EUNIS.

Class	precision	recall	f1-score	support
A3.1-A4.1	0.69	0.90	0.78	408
A3.7-A4.7	0.60	0.44	0.50	128
A5.1	0.77	0.60	0.68	125
A5.2	0.75	0.24	0.37	49
A5.4	0.86	0.29	0.44	41
avg / total	0.70	0.70	0.67	751

These results are mostly due to the absence of enough samples of all classifications to train the CNN effectively (see Figure 5) as the model is clearly overfitting as the class with more predictions is A3.1-A4.1 (see Figure 7).

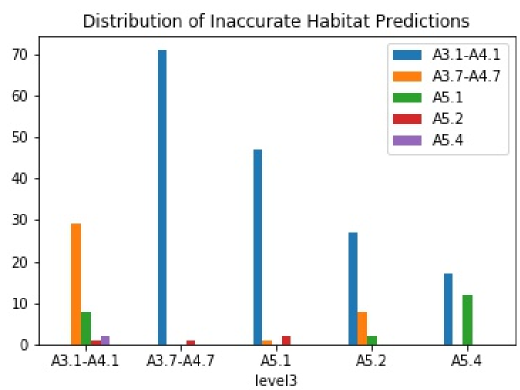


Fig. 7: Distribution of inaccurate predictions per class.

V. Conclusions and Future Work

We have successfully validated the use of AUVs for habitat mapping, by using both optic and/or sidescan sonar images collected autonomously, at a fraction of the cost of using divers or large hydrographic vessels.

The image enhancement applied to underwater optic images handled the problem associated to the lack of light in underwater environments, helping scientists to correctly classify thousands of images.

In comparison with the approaches presented in Section II (see Table V), our system performs complete habitat mapping over a vast region (80 km^2 in the case of OMARE) by using a fully autonomous system (from collection to classification). Moreover, our classification metric uses the EUNIS convention, a habitat classification scheme imposed by the European Union that aims to unify the types of identified habitats all across the world and is of practical application to both city council staff and specialists.

TABLE V: Overall accuracy comparison of related work prediction models.

Author	Best Model Accuracy
Berthold et al. [5]	56.0%
Liu et al. [4]	76.9%
Guirado et al. [7]	91.8%[1]
Gòmez-Rìos et al. [6]	98.2%[2]
Our proposal	**85.1%**

We believe that the results from CNNs are currently struggling due to the low amount of data samples currently annotated, although as per different authors in related work they tend to scale very quickly when given a significant amount of images. Also, the fact that they do not require feature engineering makes them easy to implement (using libraries such as Keras) and available to ML beginners in the scientific community of habitat mapping.

In spite of the lower prediction score of our model on the test set in comparison with the training phase score, our results are promising and, according to the results in [4], the model is expected to perform better when adding more images to the training phase.

In regards to the enhancement of underwater optic images, specialists have benefited from a color correction transformation in order to perceive key features in underwater optic images. This statement comes from their positive feedback during the annotation phase.

In the future, we intend to include more images to train the existing model, in order to make a new evaluation of the model. After that, we want to transfer the learning of the model described in this paper and proceed to classify sidescan sonar images. We also intend to predict higher levels of the EUNIS taxonomy.

References

[1] W. Laurance, "Habitat destruction: Death by a thousand cuts," 01 2010.

[2] "Omare," http://www.omare.pt/en/, 2018, [Online; Accessed 30-September-2018].

[3] L. Madureira, A. Sousa, J. Braga, P. Calado, P. Dias, R. Martins, J. Pinto, and J. Sousa, "The light autonomous underwater vehicle: Evolutions and networking," in *OCEANS-Bergen, 2013 MTS/IEEE*. IEEE, 2013, pp. 1–6.

[4] T. Liu, A. Abd-Elrahman, J. Morton, and V. L. Wilhelm, "Comparing fully convolutional networks, random forest, support vector machine, and patch-based deep convolutional neural networks for object-based wetland mapping using images from small unmanned aircraft system," *GIScience & Remote Sensing*, vol. 55, no. 2, pp. 243–264, 2018. [Online]. Available: https://doi.org/10.1080/15481603.2018.1426091

[5] T. Berthold, A. Leichter, B. Rosenhahn, V. Berkhahn, and J. Valerius, "Seabed sediment classification of side-scan sonar data using convolutional neural networks," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov 2017, pp. 1–8.

[1] Calculated by us, as the paper only had precision, recall and F1 scores available.

[2] Combined accuracy for all datasets.

[6] A. Gòmez-Rìos, S. Tabik, J. Luengo, A. Shihavuddin, B. Krawczyk, and F. Herrera, “Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation,” 03 2018.

[7] E. Guirado, S. Tabik, D. Alcaraz-Segura, J. Cabello, and F. Herrera, “Deep-learning versus obia for scattered shrub detection with google earth imagery: Ziziphus lotus as case study,” *Remote Sensing*, vol. 9, no. 12, p. 1220, 2017.

[8] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson, 2016.

[9] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann, 2017.

[10] D. Nikitenkot, M. Wirthl, and K. Trudel, “White-balancing algorithms in colour photograph restoration,” in *2007 IEEE International Conference on Systems, Man and Cybernetics*, Oct 2007, pp. 1037–1042.

[11] R. Gonzalez and R. Woods, *Digital Image Processing*. Pearson/Prentice Hall, 2008. [Online]. Available: https://books.google.pt/books?id=8uGOnjRGEzoC

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, p. 8490, 2017.

[13] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[14] S. Lawrence, C. Giles, A. C. Tsoi, and A. Back, “Face recognition: a convolutional neural-network approach,” *IEEE Transactions on Neural Networks*, vol. 8, no. 1, p. 98113, 1997.

[15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[16] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei, “Imagenet: A large-scale hierarchical image database,” in *In CVPR*, 2009.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[18] “Keras: The python deep learning library,” https://keras.io/, [Online; Accessed 27-July-2018].

[19] “Tensorflow,” https://www.tensorflow.org/, [Online; Accessed 27-July-2018].