 doi:10.20944/preprints201902.0257.v2

Technical Note

Pattern Recognition with Convolutional Neural Networks: Humpback Whale Tails

Luis Fernando de Mingo López *[1,†,‡], **Clemencio Morales Lucas** [1,†,‡], **Nuria Gómez Blas** [1,†,‡] and **Krassimira Ivanova**[2,†]

[1] Escuela Técnica Superior de Ingeniería de Sistemas Informáticos, Universidad Politécnica de Madrid, Madrid 28031, Spain; fernando.demingo@upm.es (L.F.M.L.); clemencio.morales.lucas@alumnos.upm.es (C.M.L.); nuria.gomez.blas@upm.es (N.G.B.)

[2] University of Telecommunications and Post, 1 Acad. St. Mladenov Str., Sofia 1700, Bulgaria; krasy78@mail.bg (K.I.)

* Correspondence: fernando.demingo@upm.es; Tel.: +34-91-067-3566

† These authors contributed equally to this work.

‡ Current address: Escuela Técnica Superior de Ingeniería de Sistemas Informáticos, Calle Alan Turing s/n, 28031 Madrid, Spain.

Abstract: This paper presents a study and implementation of a convolutional neural network to identify and recognize humpback whale specimens from the unique patterns of their tails. Starting from a dataset composed of images of whale tails, all the phases of the process of creation and training of a neural network are detailed – from the analysis and pre-processing of images to the elaboration of predictions, using TensorFlow and Keras frameworks. Other possible alternatives are also explained when it comes to tackling this problem and the complications that have arisen during the process of developing this paper.

Keywords: convolutional neural networks; pattern recognition; machine learning

1. Introductionn

Humpback whales have patterns of black and white pigmentation and scars on the underside of their tails that are unique to each whale, just as fingerprints are to humans (see figure 5). Researchers document the marks on the right and left lobes of the tail, or flukes, and rate the percentage of dark vs. light skin pigmentation from 100 percent white to 100 percent black.

For scientific purposes, each humpback whale sighted in the North Atlantic is assigned a catalog number. The unique scarring and shading patterns also provide the inspiration for common names. For Gulf of Maine humpbacks, researchers and naturalists work together each year to name new adult whales and young animals sighted in a second year. New calves are not named because their coloring and scarring often change dramatically during that first year.

Information collected for humpbacks in the sanctuary constitutes the longest and most detailed data set for baleen whales in the world. Photographs in the Gulf of Maine Humpback Whale Catalog, maintained by the Provincetown Center for Coastal Studies, and the North Atlantic Humpback Whale Catalog, maintained by the College of the Atlantic in Maine, allow scientists and naturalists to identify and monitor individual animals and gather valuable information about population sizes, migration, health, sexual maturity and behavior patterns. Photographing individual whales and their calves each year helps to identify family relationships. Four generations of humpback whales have been documented in certain maternal lines or matrilines.

The computing performance of Artificial Intelligence has increased remarkably in recent years. While it has been available to more and more people at the same time, its technological and social impact will grow exponentially. This fact has included Artificial Intelligence in almost any field of Information Technology, with massive companies such as Google, Facebook, Amazon or Microsoft

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2

that offer services, solutions and tools based on Artificial Intelligence such as: Video Games, Virtual Assistants and Financial Services.

One of the main areas that has the most development is the field of image/pattern recognition. This is mainly due to the utility and social precision (compared to the human eye) offered by this field. This type of technology is used in a wide range of systems, from surveillance tools and facial recognition to medical applications such as the early identification of tumors.

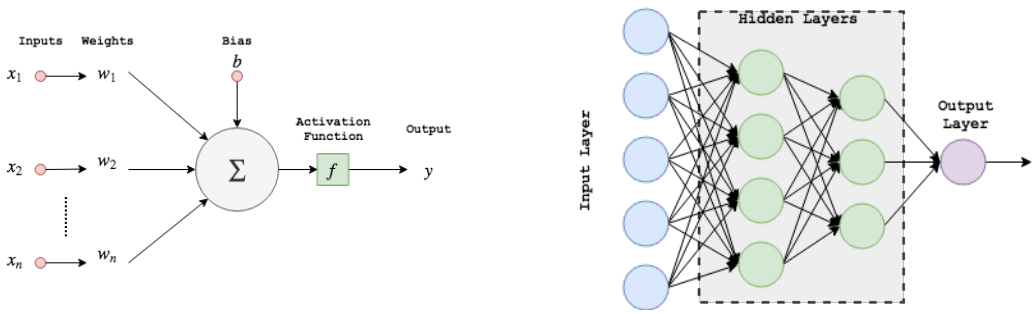
Their potential and social involvement is so high that their development must be deeply linked to ethical responsibility. It is notable that Artificial Intelligence and field recognition are not subject to controversy, as long as they receive criticism due to abusive practices or lack of ethics/privacy.

From the first years of Information Technology, the possibility that the machines were able to think has been really attractive, and has reached the minds of several writers and artists along with history. They imagined androids that were absolutely indistinguishable from humans and artificial intelligence with capabilities that the human mind is incapable of understanding, among many others. However, to understand the feasibility of these examples, it is necessary to understand what Artificial Intelligence is and how it works.

The term Artificial Intelligence has been raised historically from different points of view: the ability to think or the ability to act intelligently [1]. On the one hand, the first focuses on the approach of a human idea of intelligence, in which machines think and are rational. On the other hand, the second approach is based not so much on the process as on the result, considering the Artificial Intelligence to the ability to act and emulate what would be the result of a strictly rational action. A fundamental part of intelligence lies in learning, a process through which, through information, study and experience, a certain amount of training is achieved. That is why the need arises within the framework of Artificial Intelligence to adequately equip the knowledge systems.

With this objective, automatic learning or automatic learning was born, which, thanks to data processing, seeks to identify common patterns that allow the elaboration of increasingly precise and improved predictions. However, these algorithms have historically required complex statistical knowledge. Following the evolution of machine learning, recent years have seen the birth of a new concept, known as deep learning. Unlike machine learning, deep learning understands the world as a hierarchy of concepts [2] diluting the information in different layers through the use of modules, which transform their representation into a higher and more abstract level. This allows the amplification of the relevant information and eliminates the superfluous one [3].

Artificial neural networks or neural networks are mathematical models that try to emulate the natural behavior of biological neural networks. In these models, a network of logical units or neurons interconnected with each other is established. With this connection, they can process the received information and issue a result to the next layer determined by an activation function that takes into account the weight of each input, see figure 1b. This behavior adds more importance to specific incoming connections.



(a) Artificial neuron **(b)** Multilayer perceptron

Figure 1. Illustration of an artificial neuron and a simple neural network with 2 hidden layers. In practice, there could be many layers and many neurons per layer. Note that the output of a neuron depends on a non-linear combination of the inputs, provided $f(x)$ is a non-linear function.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2

In this model, output obtained in neuron y (figure 1a) is given by equation (1), where $\bar{x} = \{x_1, \cdots, x_n\}$ represents the input data, $\bar{w} = \{w_1, \cdots, w_n\}$ is the weight matrix and b is the so-called the bias term.

$$y = f(\sum_{i=1}^{n} w_i x_i + b) \tag{1}$$

During the training phase of a neural network the weight and bias parameters are readjusted in order to adapt the model to a specific task and improve the predictions it gets. The activation function f will be selected according to the problem to solve (a sigmoid or hyperbolic function).

Multilayer neural networks organize and group artificial neurons into levels or layers. These have an input layer and an output layer, and might have a variable number of hidden layers between them.

The input layer is formed by neurons that introduce the information into the network, but they do not produce processing, so they only act as a receiver and propagator. The hidden layers are formed by those neurons where both the entrance and the exit connect with other layers of neurons. The output layer is the last level of the network and produces a set of results out of it.

The connections of the multilayer neural networks usually go forward, connecting the neurons with their next layer. They are called feed forward networks.

2. Convolutional Neural Networks

Convolutional Neural Networks (ConvNet or CNNs) [4,5] are a class of multilayer feed forward neural networks specially designed for the recognition and classification of images.

Computers perceive images in different ways to humans, as long as for these an image consists of a two-dimensional vector with the values relative to the pixels (figure 2). They have got a channel for greyscale images or three of them for colour (RGB).

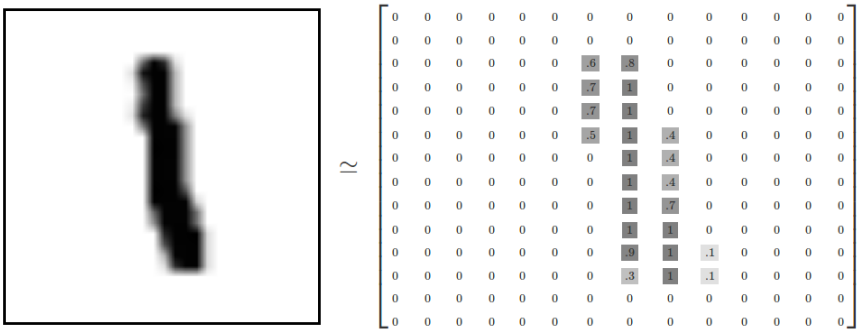


Figure 2. MNIST image vector. Left image is the picture taken and right one is the coded image using a matrix of real values $\in [0, 1]$, where 0 means a white pixel and 1 a black one.

Convolutional networks follow a certain structure, with three main types of layers: Convolutional Layer, Pooling Layer and Fully-Connected Layer, see figure 3. A series of alternate conversions and subsamples or reductions are made, until finally through a series of completely connected layers (multilayer perception) the desired output is obtained, equivalent to the number of classes.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2

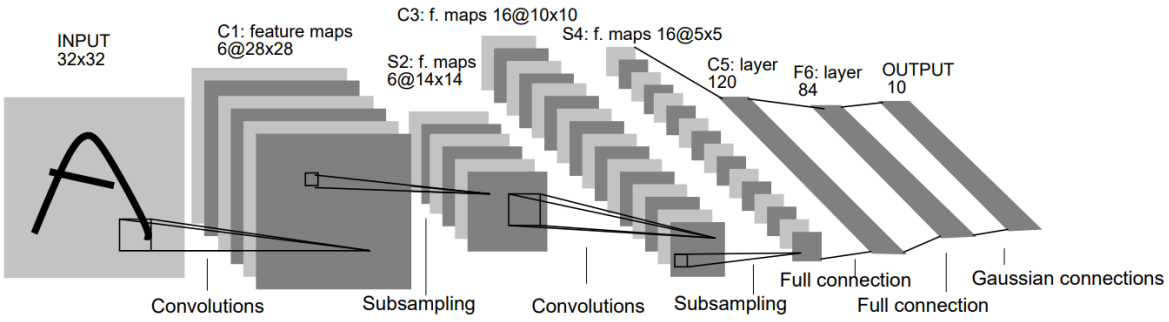
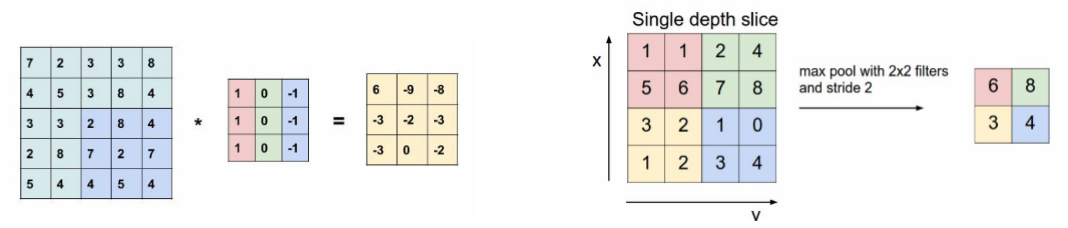


Figure 3. LeNet-5 architecture (original image published by LeCun Y. et al. [6]).

The convolution makes a series of products and sums between the starting matrix and a kernel matrix or filter of size n. On the other hand, the sub-sampling reduces the dimension of the input matrix by dividing it into sub-regions and allowing the generalization of the characteristics (figure 4).



(a) Convolution **(b)** Max Pooling

Figure 4. Matrix operators in Convolutional neural networks. Convolution involves a sum of element by element multiplication, which in turn is the same as a dot product on multidimensional matrices which machine learning researchers call tensors.

There are different architectures that are currently considered as the state of the art, such as AlexNet, Inception o VGGNet, highlighting residual neural networks o ResNet [7] among them.

2.1. Programming Language

Among the most popular programming languages in the field of Machine Learning, two of them stand out: Python and R, being the latter largely oriented towards statistical analysis. In terms of deep learning, the clear dominator is Python, thanks in part to the large number of libraries and frameworks developed for this language, such as PyTorch, Caffe, Theano, TensorFlow or Keras. That is why Python have been chosen in its version 3.6 for this project. In addition, different libraries implemented for this language will be used to facilitate the development process. Some of the most relevant packages are the following:

- NumPy, focused on the scientific computation, provides vectors or arrays as well as powerful mathematical tools on them. Pandas for the analysis of data, mainly the reading process of the different CSV files needed. Pandas allows quick access to the data, as well as a powerful treatment of it.
- Scikit-learn, a machine learning library, with powerful statistical tools that will be used mainly during the pre-processing of the data, prior to the implementation of the neural network.
- Python Imaging Library (PIL) used for reading and converting images. In the development phase, although eliminated in the final version, the Matplotlib library has been used in order to create the needed graphs. More specifically, it has been used only for visualization and to obtain the images, both original and processed, during the elaboration of this document. Therefore, it lacks relevance and usefulness in the final versions.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2

- Developed by Google in order to meet their needs along the machine learning environment, TensorFlow is a library for numerical calculations that mainly uses data flow diagrams. Published under a license of open code in 2015, it has since become one of the most popular benchmarks in the development of deep learning systems and neural networks.
- Born with flexibility and usability in mind, Keras is a library for high level neural networks that was developed for Python. One of the biggest advantages when it comes to Keras usage is that it seeks to greatly simplify the development-related tasks. Is it possible to run it with libraries such as TensorFlow, Theano or CNTK as a backend, understanding each other as an interface rather than as a framework. In 2017, Keras was integrated into the source code of TensorFlow allowing its development with a higher level of abstraction. Deep learning and its development have been greatly facilitated by the use of GPUs (Graphics Processing Unit). The high number of calculations that are carried out and their high complexity, especially during the training of a neural network, make high-performance hardware an actual need. GPUs come into play in this current scenario, as long as its usage in the training of neural networks allows to reduce considerably the time taken, compared to the exclusive use of CPUs. This is due to its high number of cores that allow parallel processing.
- CUDA, NVIDIA parallel calculus architecture, next to cuDNN, its library for deep neural networks, allows the use of GPUs in TensorFlow for our project.

The performance of the application can vary considerably depending on the available specifications, being critical in the final results and, especially, in the total execution time.

The system used has an Intel Core i7-8700 6-core processor with a base frequency of 3.2GHz up to 4.6GHz and 16GB of DDR4 RAM. It also has a NVIDIA GeForce GTX 1060 graphics card with 3GB of VDR DDR5 memory with a total of 1152 CUDA cores. This hardware is able to perform the training of the neural network and the processing of images in a comfortable way, however different approaches to the problem would require more memory allocated in the GPU.

3. Design

3.1. Dataset

The key aspect to face the construction of a neural network is the study and analysis of the dataset on which it is intended to work. Within this project, the dataset is constituted by a set of images of humpback whales, more specifically their tails, see figure 5.

(a) 1a36b244 **(b)** 1c5d333f **(c)** 1efa630a **(d)** 2c77045a

Figure 5. Humpback whale tails images in the dataset.

As the images show, the tails present different characteristics. The most notable difference a priori is the variation of colour between them, which can be presented both in greyscale or in colour. The contrast of resolutions, or the presence of elements external to the whale itself, such as annotations, is also a relevant aspect.

There are approximately 25,000 images divided into two sets, one training 9851 images while the other one evaluates 15,600 images. This distribution hinders the subsequent training of the network since the size of the training set is notably smaller to its homologous.

Alongside the images, the training set provides a CSV file that collects the labels of each image, check table 1.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2

Table 1. Header of the training file

	Image	Id
0	00022e1a.jpg	w_d15442c
1	000466c4.jpg	w_1287fbc
2	00087b01.jpg	w_da2efe0
3	001296d5.jpg	w_19e5482
4	0014cfdf.jpg	w_f22f3e3

Our neural network will deal with the following problem: Identify humpback whale specimens and if there is no record of it, catalogue it as a new whale with the label `new_whale`, counting in total with 4251 different classes or individuals.

3.2. Image Processing

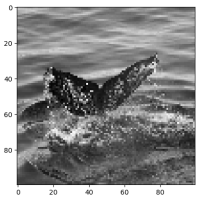
It is necessary to process each image in advance in order to facilitate the extraction of the present features present in it. If all the data is homogenized, this effect is achieved.

Firstly, the image is converted to a grey scale (figure 6), ranging from three different channel colours to a single one. There is a double reason behind this conversion – the existence of original images in the black and white dataset and the absence of colour characteristics and information. This fact provokes that only the pattern of the tail stands out as useful information.

(a) Original image **(b)** Red component **(c)** Green component **(d)** Blue component **(e)** grey scale image

Figure 6. RGB image to grey image conversion.

The neural network requires the dimension of the input vectors to be fixed, and that is why each image must be rescaled beforehand, resulting, in this case, in 100x100 size matrices with a single channel.



(a) Scaled image

182	179	179	···	102	101	104
202	202	195	···	95	99	102
172	175	179	···	114	118	122
			···			
64	64	179	···	95	100	126
64	64	64	···	30	81	114
68	72	68	···	135	106	151

(b) Matrix representation

Figure 7. Scale image process.

Each pixel has a value ranging between 0 and 255. This amplitude of range, due to the operation of the convolutional networks, allows the incorrect identification of the characteristics for each vector. To correct this disadvantage, it is convenient to normalize the image previously, in a process known as zero mean and unit variance normalization (figure 7).

The first step is to center the image on the value 0 by subtracting its mean from each value of the matrix.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2

$$M = \begin{Bmatrix} 67.8595 & 64.8595 & \cdots & -13.140503 & -10.140503 \\ 87.8595 & 87.8595 & \cdots & -15.140503 & -12.140503 \\ & & \cdots & & \\ -50.140503 & -50.140503 & \cdots & -33.140503 & -0.14050293 \\ -46.140503 & -42.140503 & \cdots & -8.140503 & 36.859497 \end{Bmatrix} \quad (2)$$

Subsequently, the range is compressed dividing each value between matrix's standard deviation.

$$M = \begin{Bmatrix} 1.4328924 & 1.3695457 & \cdots & -0.2774693 & -0.21412258 \\ 1.855204 & 1.855204 & \cdots & -0.31970045 & -0.25635374 \\ & & \cdots & & \\ -1.0587456 & -1.0587456 & \cdots & -0.6997808 & -0.0029668 \\ -0.97428334 & -0.88982105 & \cdots & -0.17189142 & 0.7783096 \end{Bmatrix} \quad (3)$$

3.3. Data Augmentation

A common practice within the classification of images through neural networks is the increase of data present in the dataset. This is especially useful where the proportion of classes is not balanced and there are numerous categories with only one sample.

In order to increase the number of data available for the training, a series of processes are performed on an original image, thus generating a series of derived images. Among the possible modifications are the rotation (figure 8), translation, noise reduction, etc.

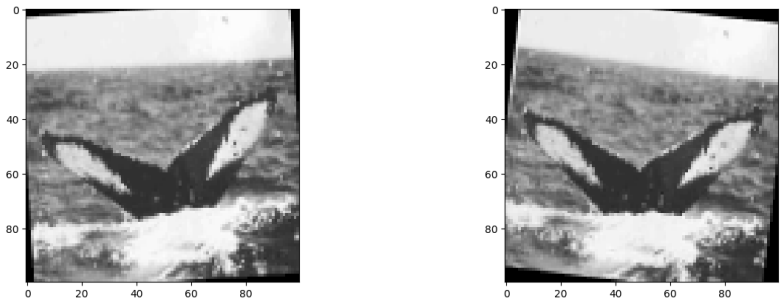


Figure 8. Image rotation.

The effectiveness of this technique lies in the way neural networks understand the images and their characteristics. If an image is slightly modified, it is perceived by the network as a completely different image belonging to the same class.

This decreases the chances of the network to focus on irrelevant orientations or positions while maintaining the relevance of the desired characteristics such as the pattern of the tail in this project.

An increase of the training set has been made following the following criteria: If a class has less than 10 images, the difference with its original sample number is generated. Therefore, if a specimen had 4 samples, an additional one would be generated from the images associated with a whale, see figure 9.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2

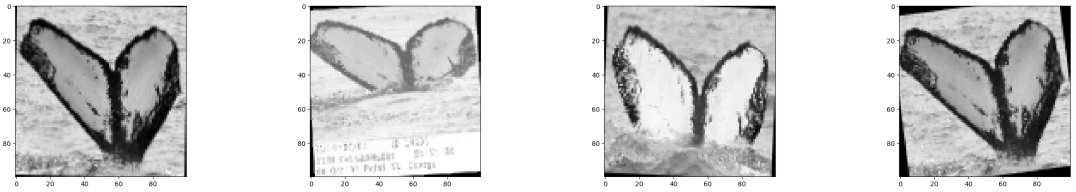


Figure 9. Augment over specimen `w_964c1b3`.

3.4. Neural network architecture

The implemented network presents structure shown in table 2.

Table 2. Architecture of the implemented network.

Layer (type)	Output Shape	Param #
conv0 (Conv2D)	(none, 94, 94, 32)	1600
bn (BatchNormalization)	(None, 94, 94, 32)	128
activation (Activation)	(None, 94, 94, 32)	0
max_pool (maxPooling2D)	(None, 47, 47, 32)	0
conv1 (Conv2D)	(None, 45, 45, 64)	18496
activation_1 (Activation)	(None, 45, 45, 64)	0
avg_pool (AveragePooling2D)	(None, 15, 15, 64)	0
flatten (Flatten)	(None, 14400)	0
ReLU (Dense)	(None, 450)	6480450
dropout (Dropout)	(None, 450)	0
softmax (Dense)	(None, 4251)	1917201
Total params: 8,417,875		
Trainable params: 8,417,811		
Non-trainable params: 64		

A convolutional layer is established, followed by batch normalization [8] and a max pool reduction. Following this fact, another convolutional layer is defined, followed by a reduction by mean or average pooling. This firstly allows extracting the most important characteristics of the image. Secondly, at the next level of depth, smoothing the extraction ensures the lack of loss of relevant information.

In both layers, a ReLU (Rectified Linear Unit) activation function is used, defined by:

$$f(x) = x^+ = \max(0, x) \tag{4}$$

Where x is the input of the neuron, returning the value x if it is positive or 0 if it is negative. This allows to considerably accelerate the training to be easily computable in comparison to other activation functions such as softmax.

Finally, a conventional multilayer neural network is connected. This network is formed by 450 neurons with dropout, followed by a dense output layer which is completely connected to softmax as an activation function, as well as 4251 neurons that are equivalent to the total number of classes to classify. The first layer receives as input a vector of one dimension, which is achieved compressing the output of two dimensions obtained from the convolutional layers.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2

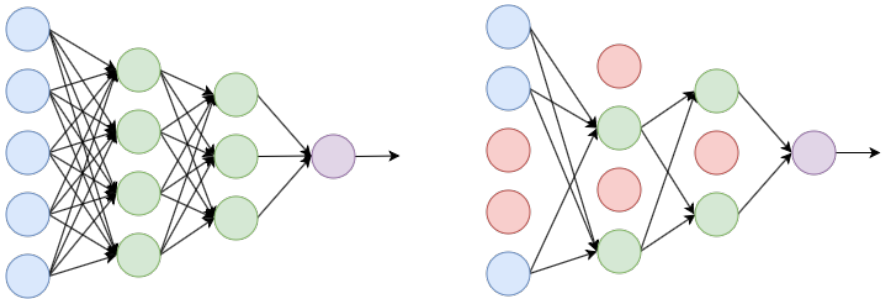


Figure 10. Dropout schema used in convolutional neural networks (Input layer in blue, hidden layers in green, output layer in purple and dropout neurons in red).

The dropout allows ignoring a percentage of input units or neurons during training, in this case 80%. This ignorance disembogues in a neural network smaller than the original and with therefore fewer parameters, which decreases the dependencies between neurons and thus avoids overtraining.

The softmax activation function is used in classification problems with multiple options, as in this case, and it returns the probabilities of each class. The aforementioned function is given by:

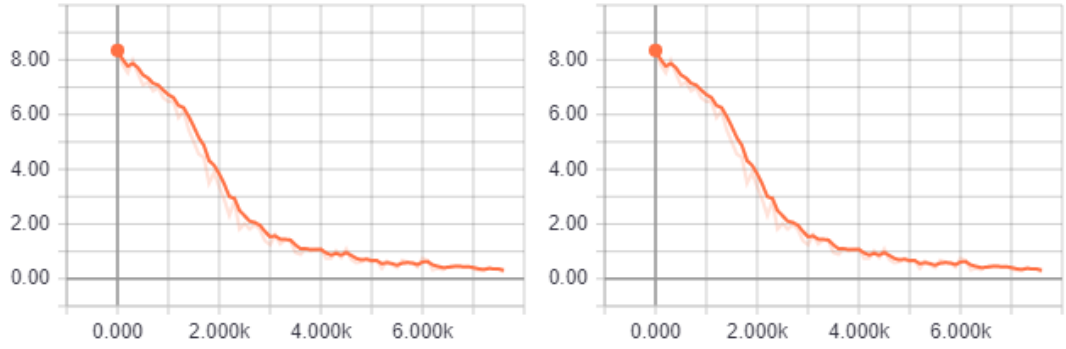
$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{4251} e^{x_j}} \tag{5}$$

It compresses the values between 0 and 1 and makes the sum of all the resulting values equal to 1, being x an input vector of size equal to the number of units or neurons of the layer (4251).

3.5. Training

TensorBoard tool (belonging TensorFlow) will be used within this section to monitor and visualize our neural network during the training phase, performing a simulation while obtaining the precision and loss graphs.

A key aspect of neural networks is the loss function that informs about how far away are the predictions of a model (y') from the real labels y. It is a positive value that improves the performance of the model while decreasing. .



(a) Loss during training **(b)** Accuracy during the training

Figure 11. Lost/Accurary during the >7000 iterations in the training process.

The loss function used on the implemented model is Categorical Cross-Entropy (CCE) [9,10], which is especially useful in problems where several excluding classes exist. CCE is preferred for training deep networks with softmax outputs, since: It puts more emphasis on correct classification and can distinguish better between "almost correctly classified" and "totally wrongly classified" than 0-1-loss, It corresponds to maximum likelihood for networks with softmax output and It has an information-theoretical interpretation based on probability and therefore makes sense to use with probabilities (which is the nature of our predictions. The function is given by:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2

$$H(y,\hat{y}) = \sum_x y_x \log \frac{1}{\hat{y_x}} = -\sum_x y_x \log \hat{y_x} \tag{6}$$

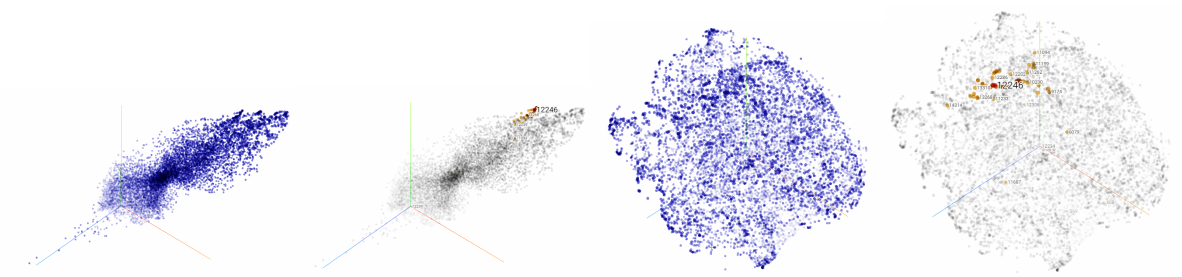
Where x is a discrete variable and $\hat{y}$ is the prediction for the real distribution y. The accuracy of the model during the training increases while the value of the loss function decreases.

The simulation of the training takes as reference the exit of the first dense layer before proceeding to the classification. This does not allow an exact representation of what would be the final output with the total classes, but easily exemplifies the behaviour of the neural network and the data during the training phase.

Four situations are differentiated during the execution in the case of study. The beginning, where the network is not trained (figure 12a). The first iterations (figure 12c) where most are considered new whale due to their common characteristics among all the images and to their greater proportion in comparison to the other classes. The phase where features from other classes are starting to be noticed (figure 13a). The final phase, where the network completes its training and differentiates with greater precision (figure 13c). A group with similar characteristics has been selected to follow its grouping.

(a) Init state **(b)** Init state **(c)** First iterations **(d)** First iterations

Figure 12. Different states in simulation.



(a) Last iterations **(b)** Last iterations **(c)** Final stage **(d)** Final state

Figure 13. Different states in simulation.

3.6. Optimization algorithm

The learning process is summarized in one idea: the minimization of the loss function by updating the parameters, w, of the network, turning it into an optimization problem.

Adam or Adaptive Moment Estimation [11] is an optimization algorithm that allows modifying the learning rate as the training is performed. Specifically, it is a variant of the stochastic gradient descent.

The neural network has a series of hyper parameters among them the learning rate α. This parameter allows the gradient descent algorithms to determine the following point, so:

$$w_j = w_j - \alpha \frac{\partial f(w_j)}{\partial w_j} \tag{7}$$

Where w_j is one of the parameters, f the loss function and α the learning rate.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2

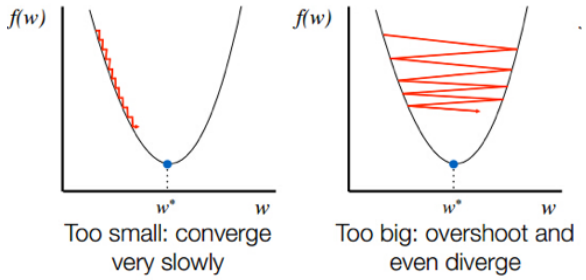


Figure 14. Learning rate α values. A small value produces slow convergence while a high value could overshoot and even diverge.

If its value is too small, the convergence, and therefore the learning, will take too long, in the same way as if the rate is sufficiently large the next point will exceed the minimum, making it impossible to reach it. It is therefore true that an adequate value of the learning rate is the key for the correct performance during the training.

The descent of the gradient is an iterative process where the weights are updated in each iteration, which is the reason why, in order to obtain the best result, it is necessary to process the dataset more than once. As long as processing a complete dataset in an iteration is difficult by memory restrictions, it is divided into parts called batches of a certain size, so if the dataset contains 1000 images and a batch size of 100, there will be 10 divisions in total.

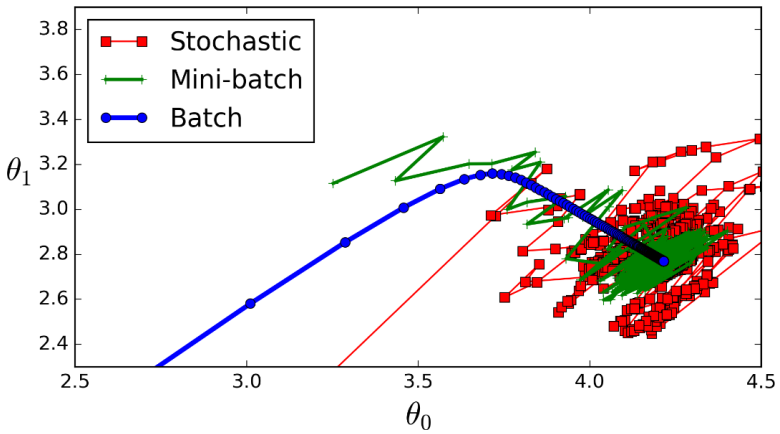


Figure 15. Learning trend using different gradient descent methods (batch, mini-batch and stochastic).

Thus, the learning process is carried out through iterations over the different batches. Depending on the size, the frequency with which the weights of the network are updated varies, so the smaller the size, the higher the frequency of update. Smaller sizes are generally used for the size of the dataset, mini-batch. If the size is equal it is called batch, and if it is equal to 1 it is known as stochastic.

In this project, a batch size of 128 over a total of 9850 images of the training set has been defined, which produces a total of 77 iterations for the achievement of the totality of the data. In addition, 100 epochs or tours on the complete dataset have been established.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2


```
4352/9850 [============>.................] - ETA: 3s - loss: 1.4541 - acc: 0.6312
4480/9850 [============>.................] - ETA: 3s - loss: 1.4526 - acc: 0.6317
4608/9850 [=============>................] - ETA: 3s - loss: 1.4529 - acc: 0.6311
4736/9850 [=============>................] - ETA: 3s - loss: 1.4621 - acc: 0.6301
4864/9850 [=============>................] - ETA: 2s - loss: 1.4667 - acc: 0.6293
4992/9850 [==============>...............] - ETA: 2s - loss: 1.4700 - acc: 0.6300
5120/9850 [==============>...............] - ETA: 2s - loss: 1.4694 - acc: 0.6314
5248/9850 [==============>...............] - ETA: 2s - loss: 1.4704 - acc: 0.6313
5376/9850 [===============>..............] - ETA: 2s - loss: 1.4741 - acc: 0.6306
5504/9850 [===============>..............] - ETA: 2s - loss: 1.4726 - acc: 0.6310
5632/9850 [===============>..............] - ETA: 2s - loss: 1.4732 - acc: 0.6300
5760/9850 [================>.............] - ETA: 2s - loss: 1.4787 - acc: 0.6280
5888/9850 [================>.............] - ETA: 2s - loss: 1.4780 - acc: 0.6279
6016/9850 [================>.............] - ETA: 2s - loss: 1.4727 - acc: 0.6287
6144/9850 [=================>............] - ETA: 2s - loss: 1.4747 - acc: 0.6273
6272/9850 [=================>............] - ETA: 2s - loss: 1.4731 - acc: 0.6276
6400/9850 [=================>............] - ETA: 2s - loss: 1.4769 - acc: 0.6272
6528/9850 [==================>...........] - ETA: 1s - loss: 1.4732 - acc: 0.6290
6656/9850 [==================>...........] - ETA: 1s - loss: 1.4765 - acc: 0.6285
6784/9850 [==================>...........] - ETA: 1s - loss: 1.4757 - acc: 0.6285
6912/9850 [===================>..........] - ETA: 1s - loss: 1.4778 - acc: 0.6282
7040/9850 [===================>..........] - ETA: 1s - loss: 1.4789 - acc: 0.6274
7168/9850 [===================>..........] - ETA: 1s - loss: 1.4786 - acc: 0.6270
7296/9850 [====================>.........] - ETA: 1s - loss: 1.4778 - acc: 0.6275
7424/9850 [====================>.........] - ETA: 1s - loss: 1.4827 - acc: 0.6270
7552/9850 [====================>.........] - ETA: 1s - loss: 1.4853 - acc: 0.6262
7680/9850 [=====================>........] - ETA: 1s - loss: 1.4851 - acc: 0.6262
7808/9850 [=====================>........] - ETA: 1s - loss: 1.4852 - acc: 0.6262
7936/9850 [=====================>........] - ETA: 1s - loss: 1.4844 - acc: 0.6256
8064/9850 [======================>.......] - ETA: 1s - loss: 1.4819 - acc: 0.6265
8192/9850 [======================>.......] - ETA: 0s - loss: 1.4787 - acc: 0.6274
8320/9850 [======================>.......] - ETA: 0s - loss: 1.4807 - acc: 0.6274
8448/9850 [=======================>......] - ETA: 0s - loss: 1.4803 - acc: 0.6274
8576/9850 [=======================>......] - ETA: 0s - loss: 1.4802 - acc: 0.6280
8704/9850 [=======================>......] - ETA: 0s - loss: 1.4784 - acc: 0.6281
8832/9850 [========================>.....] - ETA: 0s - loss: 1.4829 - acc: 0.6270
8960/9850 [========================>.....] - ETA: 0s - loss: 1.4814 - acc: 0.6272
9088/9850 [========================>.....] - ETA: 0s - loss: 1.4800 - acc: 0.6275
9216/9850 [=========================>....] - ETA: 0s - loss: 1.4792 - acc: 0.6276
9344/9850 [=========================>....] - ETA: 0s - loss: 1.4820 - acc: 0.6263
9472/9850 [=========================>....] - ETA: 0s - loss: 1.4817 - acc: 0.6261
9600/9850 [==========================>...] - ETA: 0s - loss: 1.4848 - acc: 0.6254
9728/9850 [==========================>...] - ETA: 0s - loss: 1.4832 - acc: 0.6258
9850/9850 [==============================] - 6s 590us/step - loss: 1.4813 - acc: 0.6264
Epoch 72/100
```

Figure 16. Training process using Keras framework.

The dataset subjected to an increase during the intermediate stages of development has a total of 124065 images, a massive number compared to the original 9850 images. This fact produces a total of 970 iterations per epoch, having been limited to 50.

3.7. Label coding

Due to the mathematical operations they perform, most of machine learning algorithms do not allow working on categorical data (labels), requiring numerical values.

The encoding technique known as One-Hot has been used for the implementation of the project, where a vector is generated for each category where only one of the values can be 1, being the rest 0.

The coding process involves two operations:

$$V = \begin{bmatrix} Blue & Red & Green & White & Black \end{bmatrix} \tag{8}$$

First off, the categories are converted into a whole value.

$$V = \begin{bmatrix} 0 & 3 & 4 & 1 & 2 \end{bmatrix} \tag{9}$$

Being subsequently codified as One-Hot vectors.

$$M = \begin{Bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{Bmatrix} \tag{10}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2

4. Evaluation

One of the main drawbacks during training is the overfitting or under fitting of the data. If the training data is not enough, the system will not be able to correctly recognize images. On the contrary, if a model is over trained it will increase the precision on the training set but it will be incapable of generalizing if new data is received.

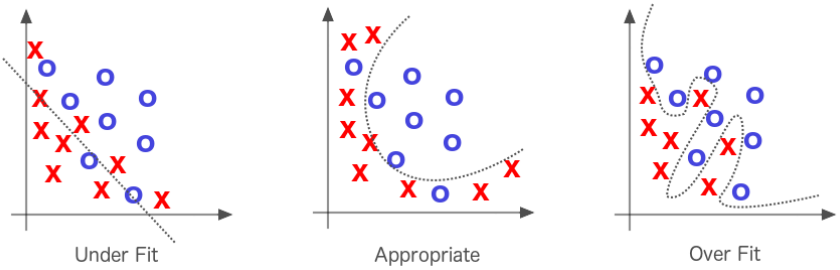


Figure 17. Model fitting.

That is why the training data is usually spited into two; a training set and an evaluation set. The test set must be representative of the total dataset data and must be large enough to be effective, usually between 70% and 80% of the original size. There may also be a third set of test to check the final version of a model, providing new data and ensuring an unbiased evaluation.

This allows to obtain results that are closer to reality since the data of the validation set remains hidden from the network during training and thus avoids overfitting the model.

In the implemented application, the validation set has been established at 10% of the total, while the training set was 90%. It has only been divided during the phase of development, by increasing the data set as described, to adjust the network's hyper parameters and study their performance, using the entire original dataset in advanced versions of the model and not the increased dataset. This is due to two factors: the considerable increase in both the time of image pre-processing and in the training time of the neural network as well as the improvement in the percentage of final success of only 1%. However, if the goal is to achieve the maximum precision, the data increase in the final version is to be incorporated, in addition to a more aggressive processing on the data.

The decisions that justify the need to establish a process of image enhancement are several. First, the asymmetrical distribution of the data set, existing categories with a single example, which would result in the presence of values exclusively in one of the sets when dividing them during the evaluation results in counterproductive. The second and most important reason is the existence of a set of test data, with unknown values, provided by Kaggle [12], allowing us to generate a CSV file with the results, upload them to the competition and compare the performance of the model. Therefore, this set of tests will be used as a validation set in the final versions of the application.

4.1. Forecasting

The whole process of designing, implementing and training a neural network is summarized in the prediction, where it is expected that the network responds with a certain precision to a series of entries.

The model implemented reaches a precision of 0.4407, which is not a probability suitable for production but due to the characteristics of the selected dataset (4251 asymmetric classes) and to the hardware limitations that could exist in terms of memory, it is considered acceptable. If we compare the public classification of the Kaggle competition, the developed model would be placed in the 54th position out of a total of 528 participants, with the highest probability being 0.78563 and the second 0.64924.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2019 doi:10.20944/preprints201902.0257.v2

To extract the results and the resulting classes it is necessary to reverse the labels coding process. In order to achieve this goal, the vector of predictions of length 4251 for each input is obtained, where a probability is established for each class. Then, the five highest probabilities are selected and their positions in the array are reverted to the original label.

5. Future improvements

5.1. Siamese Networks and Triplet Loss

The case study of this project, the identification of whales by their tail pattern where there are numerous classes belonging the same species, shares similarities with facial recognition applications. Within this field, one of the techniques with the greatest impact and performance are the Siamese networks and the use of triplet loss [13].

However, this procedure has not been implemented due to the memory restrictions of the hardware used, as long as this technique requires a large amount of memory.

Its fundamentals lie in the comparison of images to find similarities between them. Specifically, starting from an image to analyze, three elements are needed: Anchor image, where an image of the same class is obtained, positive image, and another of a different one, called negative image. The anchor image is compared with both images and an attempt is made to minimize the distance between the original image with the positive image while maximizing the distance with the negative image.

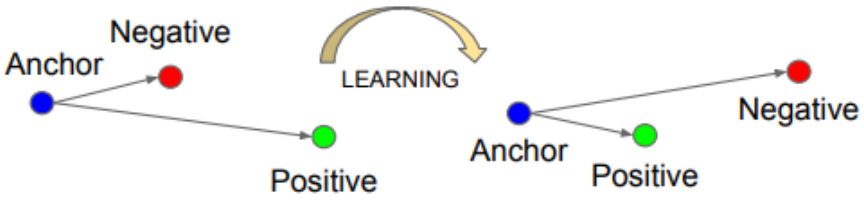


Figure 18. Triplet Loss learning.

Siamese networks are actually relevant at this stage, as long as they present an architecture that contains two or more identical subnets. In the case of triplet loss, there are three convolutional networks. These subnets share parameters that are updated simultaneously in each iteration

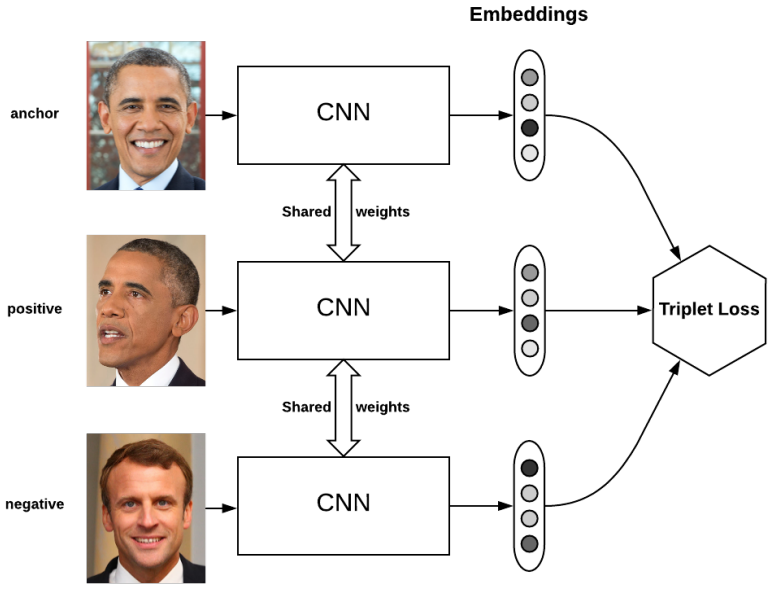


Figure 19. Triplet Loss model.

 doi:10.20944/preprints201902.0257.v2

All of the three images are analyzed, being their distances computed, while adjusting the weights according to the similarity of the images and their distances. Each network results in an interpretation of the image, process known as embedding. Therefore, when using the same model for identification, the extracted characteristics will be similar if they belong to the same class or to a different one in the opposite case.

6. Conclusions

The original approach to this project has been the study and deepening in the fields of artificial intelligence, deep learning and convolutional neural networks. That is why, despite the fact that the accuracy achieved does not imply a value close to the state of the art, the different methods and techniques used during the development of systems that allow recognizing images have been explored. Likewise, the influence and importance of previous work on the neural networks themselves has been analyzed, such as the processing of images and the comprehension of data sets.

During the development of this work, several of the original design decisions were altered. The usage of a single library for neural networks, TensorFlow, was raised, but in later versions Keras was also due to two factors: its previous inclusion in the TensorFlow library and the possibility of achieving a greater extent to offer a higher abstraction approach at a programming level compared to TensorFlow.

It has been possible to verify how the importance of the performance of a neural network does not reside exclusively in the architecture that it may have, but that the data set available substantially alters the final result, being a balanced dataset and a large number of samples a critical aspect. The main complications found are divided between the two most relevant goals: the pre-processing of images and the model. The selected dataset has greatly weighed the final accuracy of the application, requiring a much more aggressive data increase than has currently been done. Nonetheless, despite not providing the best results, it offers a clear vision about the importance of the available data, as well as the functioning of the neural networks in terms of the perception of images and how they are treated. In terms of the problems that have occurred in the implementation of the model, there are large hardware demands, especially in terms of RAM memory, that is not available in the equipment used, which has made it impossible to implement a more efficient architecture.

Author Contributions: conceptualization, C.M.L. and N.G.B.; software, L.F.d.M.L. and K.I.; investigation, C.M.L., K.I. and L.F.d.M.L.; supervision, N.G.B.; all authors analyzed the data and wrote the paper.

Funding: This research received no external funding.

Acknowledgments: This research has been partially supported by project Seguridad de Vehículos AUTOmóviles para un TRansporte Inteligente, Eficiente y Seguro. SEGVAUTO-TRIES-S2013/MIT-2713. (http://insia-upm.es/portfolio-items/proyecto-segvauto-tries-cm/?lang=en).

Conflicts of Interest: The authors declare no conflict of interest. This research has no funding sponsors so they had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Russell, S.J.; Norvig, P. *Artificial intelligence: a modern approach*; Malaysia; Pearson Education Limited,, 2016.
2. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep learning*; Vol. 1, MIT press Cambridge, 2016.
3. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature* **2015**, *521*, 436.
4. Qin, Z.; Yu, F.; Liu, C.; Chen, X. How convolutional neural network see the world - A survey of convolutional neural network visualization methods. *CoRR* **2018**, *abs/1804.11191*, [1804.11191].
5. Sewak, M.; Karim, M.R.; Pujari, P. *Practical Convolutional Neural Networks: Implement Advanced Deep Learning Models Using Python*; Packt Publishing, 2018.
6. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324.

 doi:10.20944/preprints201902.0257.v2

7. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
8. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* **2015**.
9. Theano. Categorical Cross Entropy.
10. Zhang, Z.; Sabuncu, M.R. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. *CoRR* **2018**, *abs/1805.07836*, [1805.07836].
11. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**.
12. Kaggle. Humpback Whale Identification Challenge.
13. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.