# INTRODUCTION

Plastic pollution is pollution caused by the accumulation of plastic waste in the environment. The global system of production, use and disposal of plastics is a broken system. Plastic pollution is correlated with the low cost of plastic, which leads to its massive and disposable use. It is also due to the low degradability of plastics. This pollution can have detrimental effects on land, seas and oceans, and waterways by affecting wildlife, habitat and secondarily or through feedback from humans. It is estimated that 24 to 35 million tons of plastic waste enter the aquatic environment each year. Living organisms, particularly marine animals, can be impacted, either through mechanical effects such as entanglement in plastic objects and problems associated with ingestion of plastic waste, or through exposure to chemicals in plastics that interfere with their physiology. Effects on humans include disruption of various hormonal mechanisms. In some areas, significant efforts have been made to reduce the extent of free-range plastic pollution by reducing plastic consumption, cleaning up litter, and encouraging plastic recycling.

In this paper, we will build a classification system that, based on the results obtained by analyzing a given image, will be able to say if the subject in the image is a plastic or fish. We will used tensorflow to build this model, process the data, label the data and train/test these data

# METHODOLOGY AND IMPLEMENTATION

To successfully achieve this project, we followed a methodology based on four (06) mainstages:

* The definition and understanding of the problem
* Data collection and Data preprocessing
* Automated labeling data by tensorflow
* Model conception using machine learning, deep learning and convulsional neural network
* Training and testing the model
* Result

Schema

## Definition and understanding the problem

There are many African countries affected by the plastic pollution particularly Senegal, Nigeria, Ghana, South Africa, Kenya and Cameroon.

Here’s the summary of some articles which allows to understand better the situation:

[Jeune Afrique]

The plastic occupies the first place among the sources of pollution and waste. Every year, humans produce 300 million tons of plastic garbage, and 11 million of them end up in the oceans. end up in the oceans. (UNEP) Without rapid action on a large scale, these volumes triple by 2040. The durability and low cost of this material make it attractive to industries. But this resistance to time also makes it harmful to our health. The plastic never really disappears. It turns into fine particles, which are swallowed by fish or farm animals and eventually consumed by humans through their food and tap water. This leads to an increased risk of cancer and hormonal problems.

[Greenpeace]

Scientists have identified 700 marine species affected by plastic pollution in our oceans. 9 out of 10 seabirds, 1 out of 3 sea turtles and more than half of the whale and dolphin species have ingested plastic. In the African context, we of plastic that the Nile and Niger rivers carry to the depths of the oceans.

* About 80% of the debris comes from land-based activities, with the remaining 20% coming from oil rigs at sea and large ships that directly dump or lose debris into the water.
* The ocean provides the primary source of protein for more than one billion people.
* Pollution threatens the job security of workers in the artisanal fishing sector.
* A country like Senegal has as its main source of income the fishing industry.

[Greenpeace]

In Cameroon, flooding is a recurring problem in my city, Douala. Every year, during the rainy season, we experience flooding of houses and stores, as well as sidewalks. All this affects the quality of our water and food, and has ruined the lives of many people who lose their homes, livelihoods and even family members. Plastic waste pollution does not only affect the mainland, but also the banks of the Wouri River. Six million tons of waste are produced every year by Cameroon, including 600 000 tons of plastic waste. This waste is often non-biodegradable and pollutes the environment. The quantity of fish is forced to decrease because of the plastic waste that floods the river which suffocates the mangrove," explains François Dikoume, president of the Matanda Ecotour association. "If we don't do anything today, the mangrove will disappear, the fish will disappear and if these two elements disappear, it is the river that dies."

Consequences of plastics in the sea:

* Strangulation of marine animals;
* Waste that has not sunk or been washed ashore, drifts and drift and wash up on our beaches. In addition to constituting a major ecological challenge, they are a visual pollution with real impacts on the local economy. It is thus all the tourist activity that can suffer the consequences: a decrease in the number of tourists, negative image of the town, etc.

To solve this problem, we are going to build a model able to recognize fish and plastic. Soon, this model could be implement into a robot which will be able to clean water environment (sea, ocean,) by collecting the plastic waste and then, concretely reduce the plastic pollution.

## Data collection and data preprocessing

* Collection of data is one of the major and most important tasks of any machine learning projects. Because the input we feed to the algorithms is data. So, the algorithms efficiency and accuracy depend upon the correctness and quality of data collected. So as the data same will be the output. For our “Fish or plastic” model classification, two major parameters are required: fish images and plastic images. Therefore, for this project:
* Almost 10% of both type of images (fish and plastic) has been selected manually on google, pixabay, unsplash.
* 55% of fish has been taken into the dataset “Fish Species Image Data” on Kaggle (<https://www.kaggle.com/datasets/sripaadsrinivasan/fish-species-image-data>) and 40% from the dataset “Blue Bot Dataset: Parrot fish” (https://www.kaggle.com/datasets/hiyaro/parrotfish)
* 95% of plastic images is from the drive “DeepPlastic” (<https://drive.google.com/drive/folders/1fsS_u2QpbRGynYkP6-D6cfvq8r0hpjXI>)
* Collecting the data is one task and making that data useful is an-an-another  
  vital task. Data collected from various means will be in an unorganized format and there  
  may be lot of null values, in-valid data values and unwanted data. Cleaning all these data,  
  replacing them with appropriate or approximate data, removing null and missing data, and  
  replacing them with some fixed alternate values are the basic steps in preprocessing of data.

## Automated labeling data by tensorflow

Tensorflow allows to label data automatically. For example, for our project, since it is an image classification, we had two classes. A plastic class and a fish class. To label our images, we simply created two folders with the exact names of our classes ("fish" and "plastic") and in each of these folders we respectively put the corresponding images. Tensorflow took care of labeling each image with the name of the folder where the image is located. To check the classes, we used the function:

tf.keras.utils.image\_dataset\_from\_directory(

data\_directory,

validation\_split=percentage\_of\_validation\_split,

subset="training",

seed=123,

image\_size=(img\_height, img\_width),

batch\_size=batch\_size).class\_names

## Model conception using machine learning, deep learning and convulsional neural network

### Libraries imprting

Firstly, we have imported some libraries and printed the tensorflow version:

import numpy as np

import os

import PIL

import PIL.Image

import tensorflow as tf

print(tf.\_\_version\_\_)

### Chekup

We verified if we had the correct path then we defined directories for plastic and fish folders. After, we saw what the filenames look like in the `plastic` and `fishes` training directories. And also find out the total number of plastic and fishes images in the directories.

#for showing the number of images on the directory

tf.keras.utils.image\_dataset\_from\_directory(

   "fish-or-plastic"

)

# Directory with our training plastic pictures

train\_plastic\_dir = os.path.join("fish-or-plastic/plastic")

# Directory with our training fishes pictures

train\_fish\_dir = os.path.join("fish-or-plastic/fish")

train\_plastic\_names = os.listdir(train\_plastic\_dir)

print(train\_plastic\_names[:10])

train\_fish\_names = os.listdir(train\_fish\_dir)

print(train\_fish\_names[:10])

print('total training plastic images:', len(os.listdir(train\_plastic\_dir)))

print('total training fish images:', len(os.listdir(train\_fish\_dir)))

### Visualization of images

We have used matplotlib for visualizing some images in our dataset

%matplotlib inline

import matplotlib.pyplot as plt

import matplotlib.image as mpimg

# Parameters for our graph; we'll output images in a 4x4 configuration

nrows = 4

ncols = 4

# Index for iterating over images

pic\_index = 0

# Set up matplotlib fig, and size it to fit 4x4 pics

fig = plt.gcf()

fig.set\_size\_inches(ncols \* 4, nrows \* 4)

pic\_index += 8

next\_plastic\_pix = [os.path.join(train\_plastic\_dir, fname)

                for fname in train\_plastic\_names[pic\_index-8:pic\_index]]

next\_fish\_pix = [os.path.join(train\_fish\_dir, fname)

                for fname in train\_fish\_names[pic\_index-8:pic\_index]]

for i, img\_path in enumerate(next\_plastic\_pix+next\_fish\_pix):

  # Set up subplot; subplot indices start at 1

  sp = plt.subplot(nrows, ncols, i + 1)

  sp.axis('Off') # Don't show axes (or gridlines)

  img = mpimg.imread(img\_path)

  plt.imshow(img)

plt.show()

### Data processing

We have defined the train dataset and the test dataset, and set in the both case validation split to 20%

train\_ds = tf.keras.utils.image\_dataset\_from\_directory(

  "fish-or-plastic",

  validation\_split=0.2,

  subset="training",

  seed=123,

  image\_size=(80, 80),

  batch\_size=20)

val\_ds = tf.keras.utils.image\_dataset\_from\_directory(

  "fish-or-plastic",

  validation\_split=0.2,

  subset="validation",

  seed=123,

  image\_size=(80, 80),

  batch\_size=20)

### Callback

For the resources optimization, we have created a callback which was supposed to stop training when the accuracy will be > to 97%

class myCallback(tf.keras.callbacks.Callback):

  def on\_epoch\_end(self, epoch, logs={}):

    # Check accuracy

    if(logs.get('accuracy') > 0.975):

      # Stop if threshold is met

      print("\nAccuracy is about 97% so cancelling training!")

      self.model.stop\_training = True

# Instantiate class

callbacks = myCallback()

### Defining the model and compile the model

We have created our model using one rescaling layer, four covulsional layers, four maxpooling layer, one flatten layer and two dense layers.

We compiled our moedel using the optimizer adam and SparseCategoricalCrossentropy as our loss function and used accuracy as metric

### Training and testing the model

Finally, after processing of data and training the next task is obviously testing. This is where performance of the algorithm, quality of data, and required output all appears out. From the huge dataset, collected 80 percent of the data is used for training and 20 percent of the data is reserved for testing. Training as discussed before is the process of making the machine to learn and giving it the capability to make further predictions based on the training it took. Whereas testing means already having a predefined data set with output also previously labelled and the model is tested whether it is working properly or not and is giving the right prediction or not. If maximum number of predictions are right then model will have a good accuracy percentage and is reliable to continue with otherwise better to change the model. Also, further new set of inputs and the predictions made by the model will be keep on adding to the dataset, which makes dataset more powerful and accurate. We have trained our model on 30 epochs with 8 steps per epochs.

### Results

After training and testing steps, the result was pretty interesting because the model has reach 98% accuracy both on training and validation datasets.