Understanding Clouds from Satellite Images Dataset First Week Work Report

SUBMITTED BY:	HAMZA RAQEEB	ID:	21031287

DATASET

The idea is to identify and classify distinct cloud formations in satellite imagery, and afterwards segment them accordingly. The cloud formations are categorized as Fish, Flower, Gravel, and Sugar. The competition employed images obtained from NASA Worldview, which encompassed three specific regions measuring 21 degrees in longitude and 14 degrees in latitude. The true-colour images presented here are obtained from TERRA and AQUA satellites, which orbit each region once daily [1]. In certain instances, images can be generated by merging data from two satellite orbits, with any remaining uncovered regions designated as black.

SATELLITE IMAGES CAPTURING CLOUD FORMATIONS THE CONTEXT OF ECOLOGY.

Cloud formations play a crucial role in Earth's climate system, influencing the balance of energy and offering valuable insights into the dynamics of climate. The utilization of satellite imagery and precise cloud segmentation allows for the monitoring of ecosystem health, evaluation of ecological impacts, enhancement of ecological modeling, and improved management of ecosystems. Additionally, it aids in understanding the effects of climate change on biodiversity [2]. This dataset enhances understanding of cloud-climate interactions and their ecological impacts, benefiting climate research, ecosystem monitoring, and environmental management strategies.

IMPORTANT DEPENDENCIES: Are loaded, including numpy, pandas, matplotlib.pyplot, seaborn, and cv2 (OpenCV). These dependencies are essential for various data processing, visualization, and image manipulation tasks in the project, enabling efficient data analysis and image handling within the code.

PERCENTAGE OF NULL VALUES: The percentage of null values in the 'EncodedPixels' column of the DataFrame is calculated and visualized using a pie chart. This provides an understanding of the distribution of empty and non-empty values, helping to assess the availability and completeness of the 'EncodedPixels' data for further analysis and modeling. Fig 1. Show a diagram of empty values and Non-Null values.
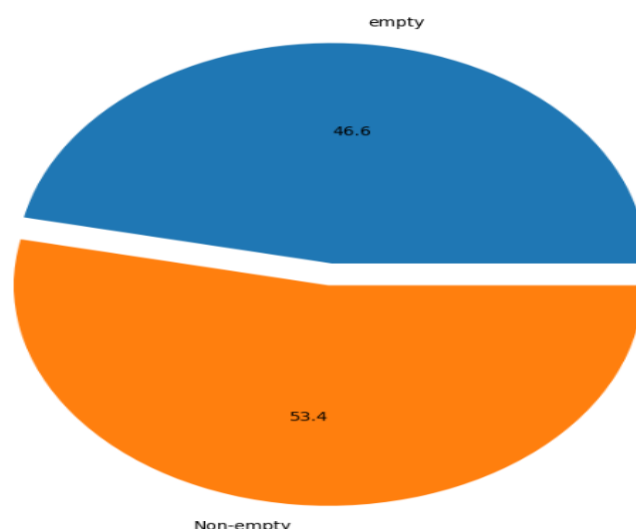

Fig. 1 Show diagram of empty and non-empty values

DATA FRAME

The resulting data frame is then displayed, showing the first few rows with the added label information for each image. This code facilitates organizing and categorizing the data based on the cloud formation labels associated with each image. Fig 2. Shows in the diagram of the data frame

```
# creating a new columns with label
df["Label"] = df["Image_Label"].apply(lambda x: x.split("_")[1])
df.head()
```

	Image_Label	EncodedPixels	Label
0	0011165.jpg_Fish	264918 937 266318 937 267718 937 269118 937 27...	Fish
1	0011165.jpg_Flower	1355565 1002 1356965 1002 1358365 1002 1359765...	Flower
2	0011165.jpg_Gravel	-1	Gravel
3	0011165.jpg_Sugar	-1	Sugar
4	002be4f.jpg_Fish	233813 878 235213 878 236613 878 238010 881 23...	Fish

Fig 2. Show diagram of Data frame

CREATE A FUNCTION THAT COUNTS THE NUMBER OF OCCURRENCE

A function that counts the number of occurrences of each label in a DataFrame column called "Label" based on specific conditions. It then generates a bar plot showing the counts of each label. Fig.3 Shows the result.
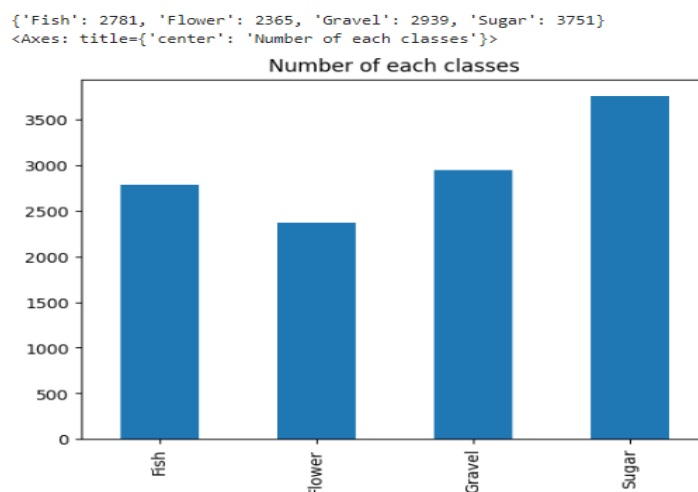

{'Fish': 2781, 'Flower': 2365, 'Gravel': 2939, 'Sugar': 3751}
<Axes: title={'center': 'Number of each classes'}>

Fig.3 Shows the numbers of each class.

CREATE A FUNCTION CALLED `DUMMY_VAR`

Defines a function called `dummy_var` that creates a dummy variable for a specific label in the DataFrame `df`. It returns a list of binary values (1 if the condition is met, 0 otherwise). The code then creates a new DataFrame called `df_images`, which contains unique image names from `df` as the "Image" column and dummy variables for each unique label in separate columns. Fig.4 shows the result.

|	Image	Fish	Flower	Gravel	Sugar
0	0011165.jpg	1	1	0	0
1	002be4f.jpg	1	1	0	1
2	0031ae9.jpg	1	1	0	1
3	0035239.jpg	0	1	1	0
4	003994e.jpg	1	0	1	1

Fig.4 Showing the results

REARRANGES THE DATA

Creates a new DataFrame called train_df by pivoting the original DataFrame df. It rearranges the data so that each unique image name becomes a row, and the minimum value of the corresponding "EncodedPixels" for each label becomes a separate column in train_df, with specific column names ('Fish_mask', 'Flower_mask', 'Gravel_mask', 'Sugar_mask') indicating the corresponding mask for each label. Fig 5 Shows the results.

|	image	Fish_mask	Flower_mask	Gravel_mask	Sugar_mask
0	0011165.jpg	264918 937 266318 937 267718 937 269118 937 27...	1355565 1002 1356965 1002 1358365 1002 1359765...	-1	-1
1	002be4f.jpg	233813 878 235213 878 236613 878 238010 881 23...	1339279 519 1340679 519 1342079 519 1343479 51...	-1	67495 350 68895 350 70295 350 71695 350 73095 ...
2	0031ae9.jpg	3510 690 4910 690 6310 690 7710 690 9110 690 1...	2047 703 3447 703 4847 703 6247 703 7647 703 9...	-1	658170 388 659570 388 660970 388 662370 388 66...
3	0035239.jpg	-1	100812 462 102212 462 103612 462 105012 462 10...	65400 380 66800 380 68200 380 69600 380 71000 ...	-1
4	003994e.jpg	2367966 18 2367985 2 2367993 8 2368002 62 2369...	-1	353317 416 354717 416 356117 416 357517 416 35...	28011 489 29411 489 30811 489 32211 489 33611 ...

Fig.5 Showing the results rearranges the data

FUNCTION DECODER

function decodes a compressed pixel representation, where consecutive pairs of numbers represent starting indices and lengths of segmented pixels. It reconstructs the segmented pixels into a 2D image by setting the corresponding pixels in a blank image array and returns the resulting image Fig.6 Shows the result.

Fig.6 Showing compressed pixel representation.
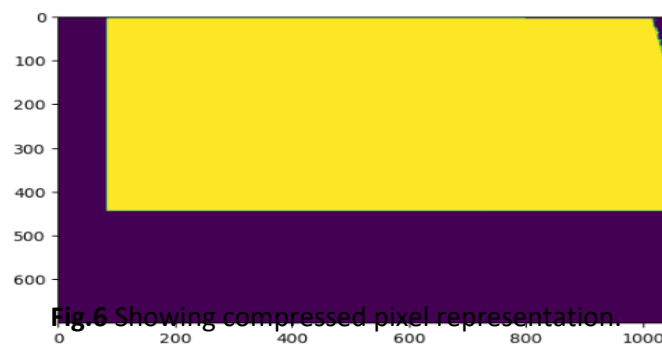
VISUALIZES SEGMENTED REGIONS

visualizes segmented regions by plotting them as images. It iterates over a subset of the dataset, finds the first non-empty encoded pixel representation, decodes it using the decode pixels' function, and displays the resulting image alongside its corresponding image name and label Fig.7 Showing the result
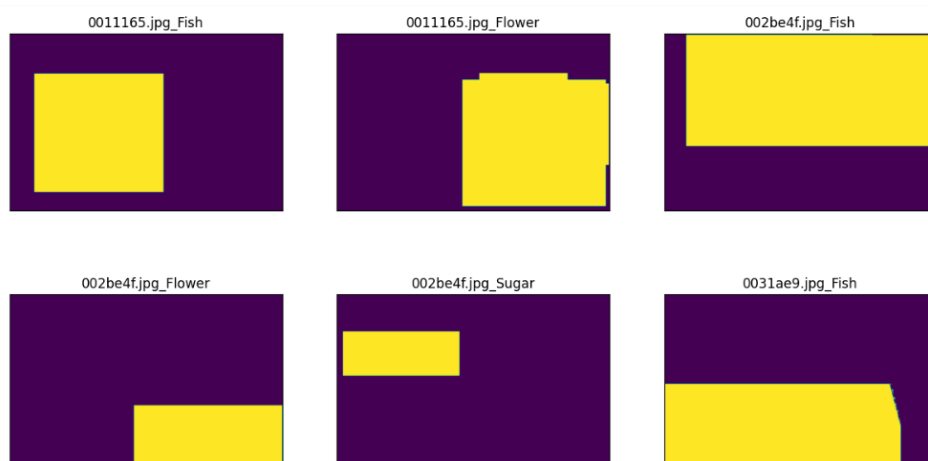
Fig 7. VISUALIZES SEGMENTED REGIONS

Proposed Model

The efficient model is a machine learning model that obtains high performance by effectively balancing accuracy and computational efficiency. It is particularly well-suited for complex tasks with limited resources, such as classifying cloud organization patterns in satellite images [3].

Why I Selected the Model

The efficient model is most effective for the provided dataset as it illustrates the effective classification of cloud organization patterns from satellite images. Its capacity to effectively manage intricate and ambiguous boundaries among various types of cloud organizations renders it well-suited for this undertaking [4]. The model's precise classification of cloud patterns enhances our comprehension of the influence of clouds on climate and aids in the advancement of dependable climate models, thereby diminishing uncertainties in future climate projections.

GOOGLE COLAB LINK:

https://colab.research.google.com/drive/1hYJro2tyWdxaytN8gyJl6e03L3zDKKgZ?usp=sharing

References:

1) Ahmed, T., & Sabab, N. H. N. (2022). Classification and understanding of cloud structures via satellite images with EfficientUNet. *SN Computer Science*, *3*, 1-11.
2) Mahmood, R., Pielke Sr, R. A., Hubbard, K. G., Niyogi, D., Dirmeyer, P. A., McAlpine, C., ... & Fall, S. (2014). Land cover changes and their biogeophysical effects on climate. *International journal of climatology*, *34*(4), 929-953.
3) Ławryńczuk, M. (2014). Computationally efficient model predictive control algorithms. *A Neural Network Approach, Studies in Systems, Decision and Control*, *3*.
4) Yuan, Y., & Hu, X. (2015). Bag-of-words and object-based classification for cloud extraction from satellite imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *8*(8), 4197-4205.