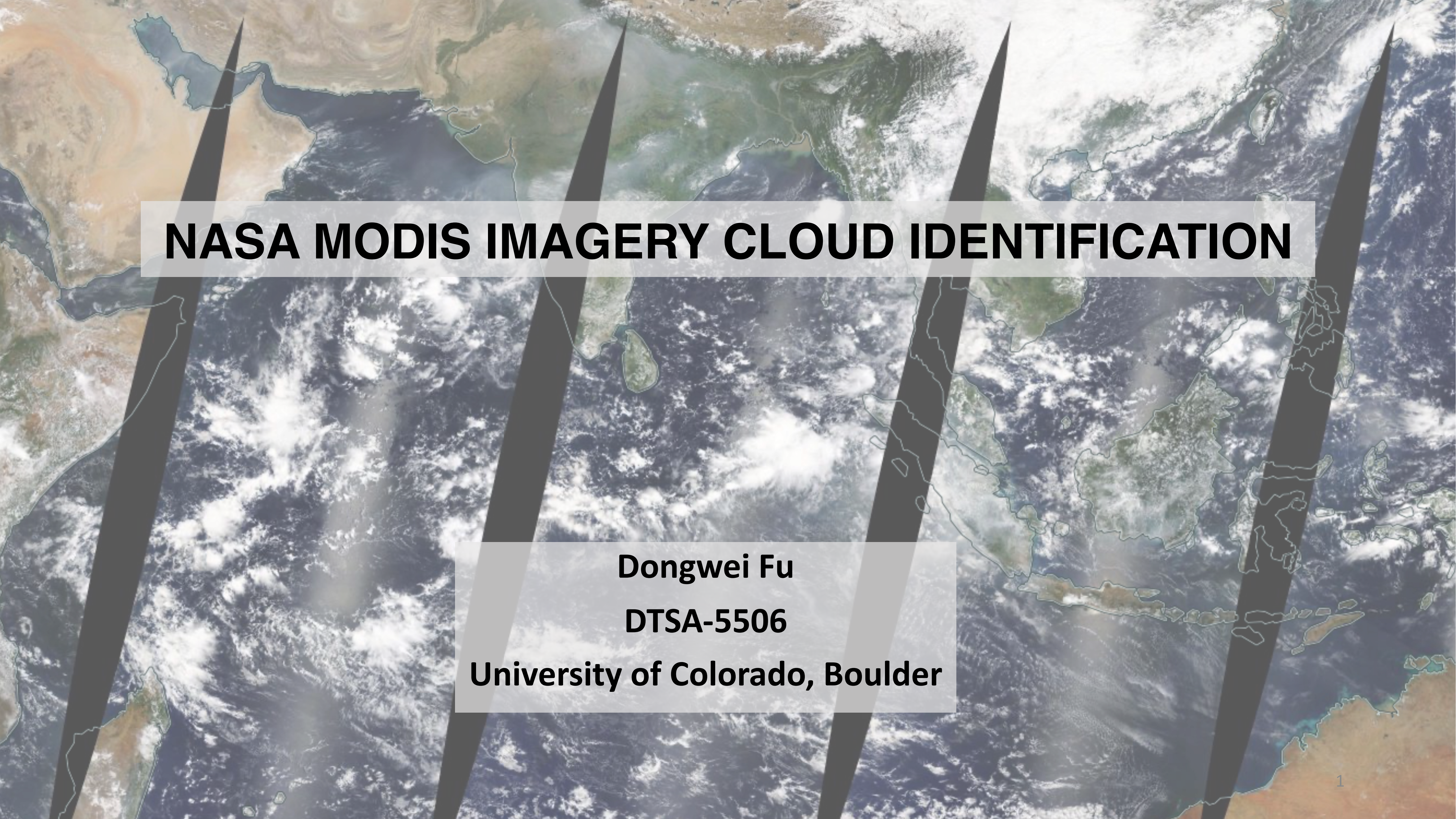


NASA MODIS IMAGERY CLOUD IDENTIFICATION

Dongwei Fu

DTSA-5506

University of Colorado, Boulder

Outline

- Problem Statement
- Related work
- Proposed work
- Timeline

Problem statement

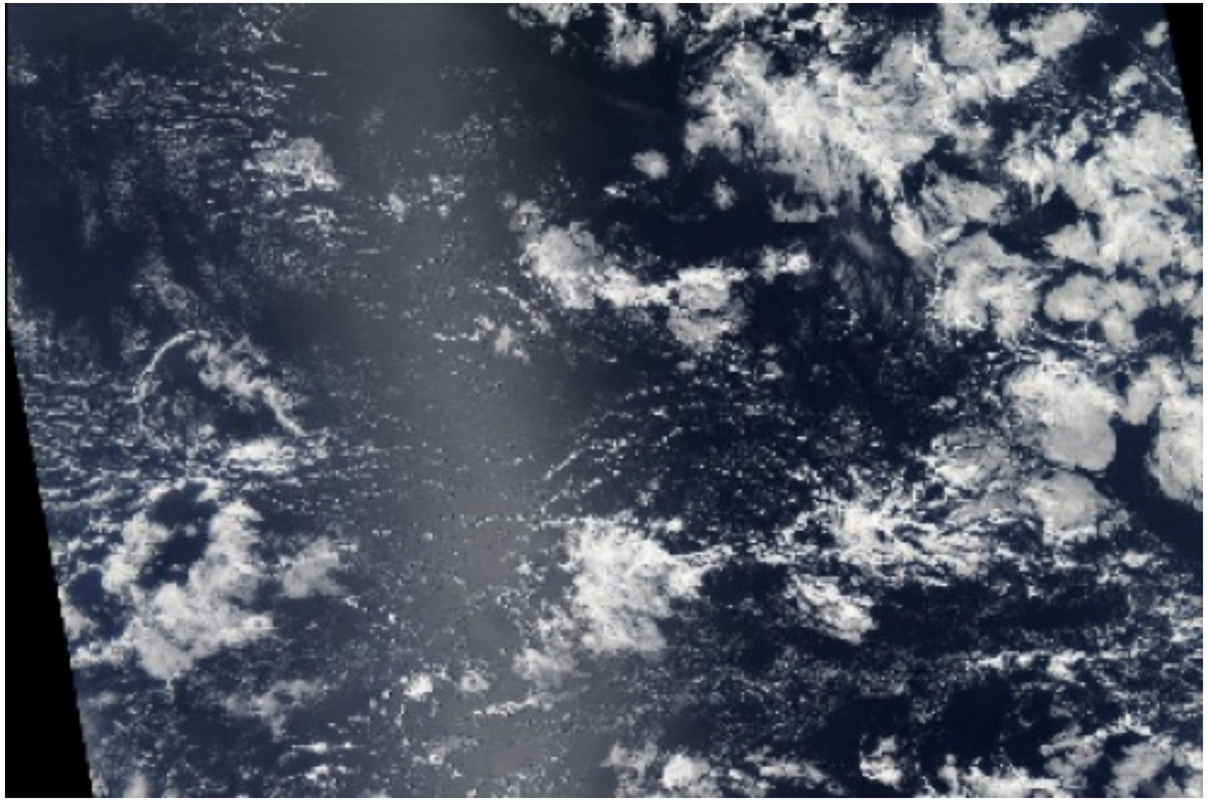
Clouds play a critical role in regulating **Earth's climate** by reflecting sunlight and trapping heat.

Small changes in **shallow clouds** properties significantly impact the Earth's climate.

Understanding shallow clouds is essential for **improving climate model predictions** of future climate.

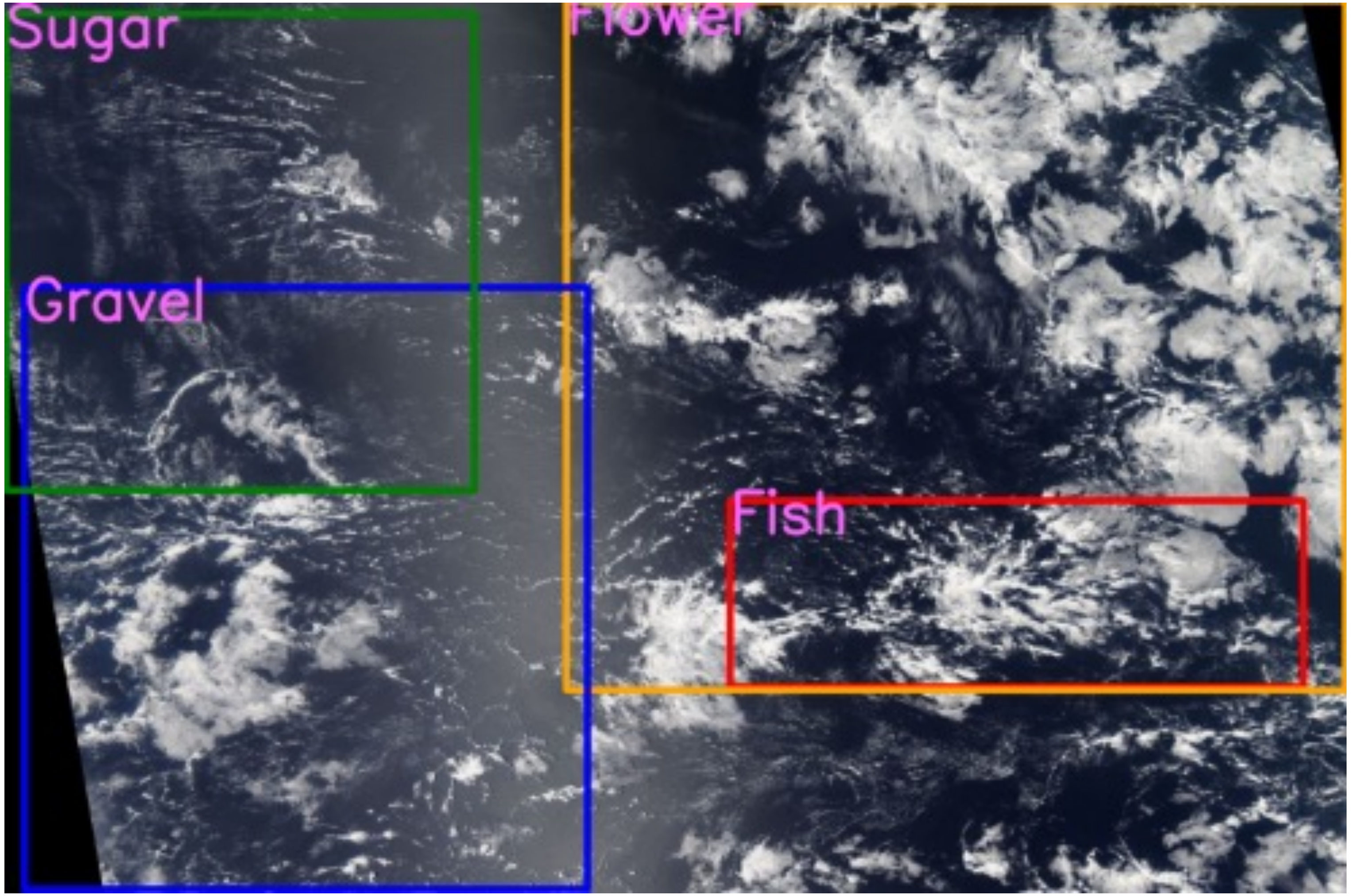
Satellite Imagery provides **extensive spatial** and **temporal coverage** of **clouds,** especially in remote regions.

Goal: Develop a Convolutional Neural Network (CNN) to identify various satellite cloud structures.



NASA Aqua MODIS 250m RGB image, from NASA WorldView.

Related Work: Satellite image dataset



Training dataset image '015aa06.jpg'

Kaggle competition dataset (Rasp et al. 2019)
Human labeled MODIS RGB image from NASA worldview web-interface. (https://worldview.earthdata.nasa.gov)

Consists of four cloud classes:

Sugar: dusting of fine clouds with little self-organization

Flower: large-scale stratiform clouds in bouquets with separations form each other.

Fish: large-scale skeletal networks that are separated from other cloud formations.

Gravel: arcs of randomly interacting cells with granularity.

(Rasp et al. 2020)

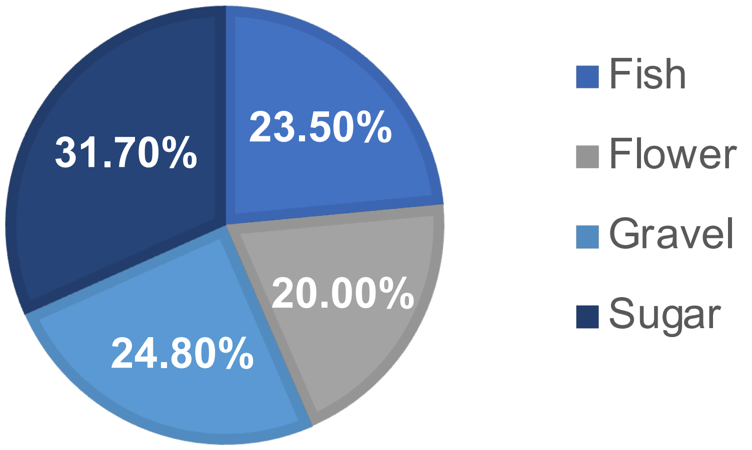
Related Work: Exploratory Data Analysis

Training dataset: 5546 RGB images
Test dataset: 3698 RGB images.
4 cloud classes:
Sugar, Flower, Fish, Gravel.

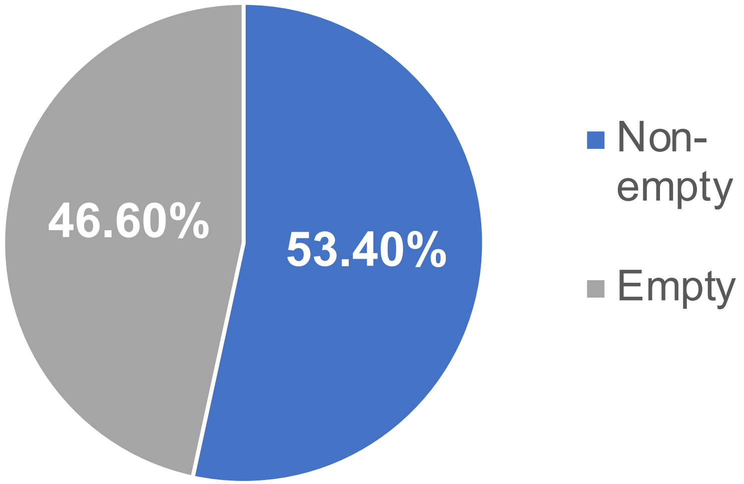
```
1  train_df.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5546 entries, 0 to 5545
Data columns (total 7 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Image_Id             5546 non-null   object
 1   Label_EncodedPixels  5546 non-null   object
 2   Fish                 5546 non-null   int64
 3   Flower               5546 non-null   int64
 4   Gravel               5546 non-null   int64
```

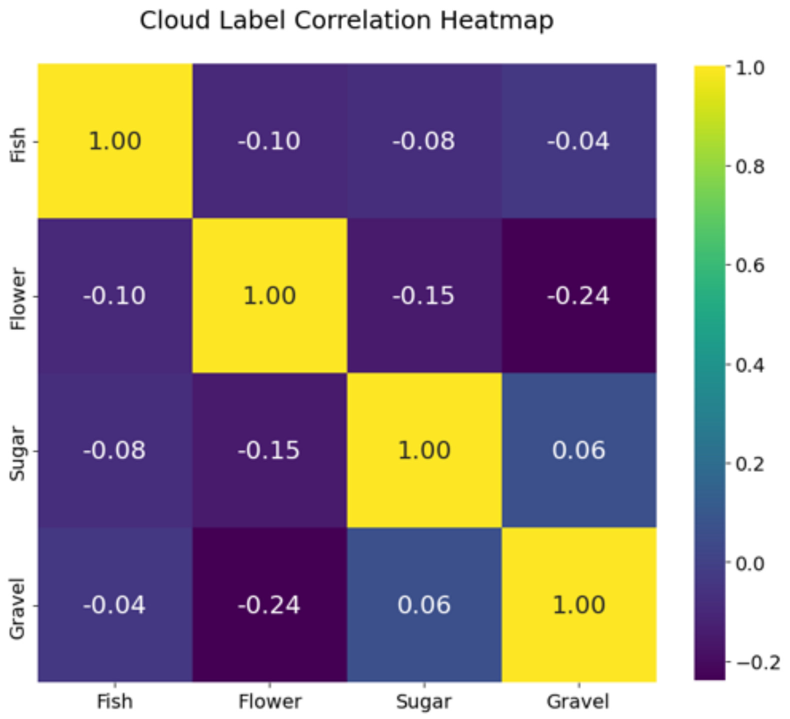
```
1  test_df.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3698 entries, 0 to 3697
Data columns (total 2 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Image_Id             3698 non-null   object
 1   Label_EncodedPixels  3698 non-null   object
dtypes: object(2)
memory usage: 57.9+ KB
```

Training label relative frequency



Training label valid mask frequency





Related Work: Data preprocessing

- **Data augmentation:** Images at 1400 x 2100 pixels resolution can be computational for heavy model training. To improve training efficiency, data augmentation such as split data into batches, perform image resizing, image flipping and random rotation.

- **Run-length encoding** (**RLE**): To reduce submission file sizes, pixel run-length encoding was implemented to record start position and run length of the masked image pixels.

Related Work: Evaluation Metrics

Dice Coefficient: $\frac{2 * |X \cap Y|}{|X| + |Y|}$

- Dice coefficient Is used to compare the **pixel-wise agreement** between a predicted segmentation and its corresponding ground truth.

Loss: $Loss$ = - $(Loss_{BCE}(yt, yp) + \ Loss_{dice}(yt, yp))$

- Loss is defined as the combination of binary cross entropy loss and dice coefficient loss.

Proposed work

1. Identify the problem and gather information on project feasibility.
2. Download data and perform EDA analysis
3. Preprocess data for segmentation model training. ***Current**
4. Build initial baseline model and assess model performance.
5. Fine-tune model / hyperparameter tuning.
6. Finalize best model.
7. Analyze final results.
8. Reach final evaluation/conclusions.

Proposed project timeline

Timeline	9/1 to 9/15	9/16 to 9/30	10/1 to 10/10	10/10 to 10/15
Action items	1. Identify problem and choose dataset. 2. EDA analysis, data	Data preprocessing 1.Data augmentation. 2.Data RLE encoding. 3. Data visualization.	1. Select model architecture and build baseline model. 2. Select alternative model and compare model performance.	1. Hyperparameter tuning to improve model performance. 2. Finalize analysis. 3. Write project report and presentation.