Weather forecast using cloud images

Present to

Punnarai Siricharoen, Ph.D

By

Peeranath Theerawatanachai 6231343821

Vorapon Khunakornkorbkij 6231356021

2110431 Intro to Digital Imaging, Semester 1/2021

Department of Computer Engineering,

Faculty of Engineering, Chulalongkorn University

Table of contents

1. Overview

1.1 Objective

To forecast the weather in a certain location at a specific time using an image of the sky that should contain no other things but the sky and clouds.

1.2 Motivation

The current weather forecasting system primarily relies on satellite pictures to forecast the weather. The majority of websites that provide weather forecasts use GPS and satellite image data to anticipate the weather. We rarely come across applications or websites that use digital image processing to predict weather.

Every person can acquire the weather simply by snapping a single digital photo of the cloud and sky in front of them.

2. Image Dataset

- Training: SWIMSEG dataset, which includes 1013 cloud images (many kinds of cloud)

Reference:

https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/HEJTK1

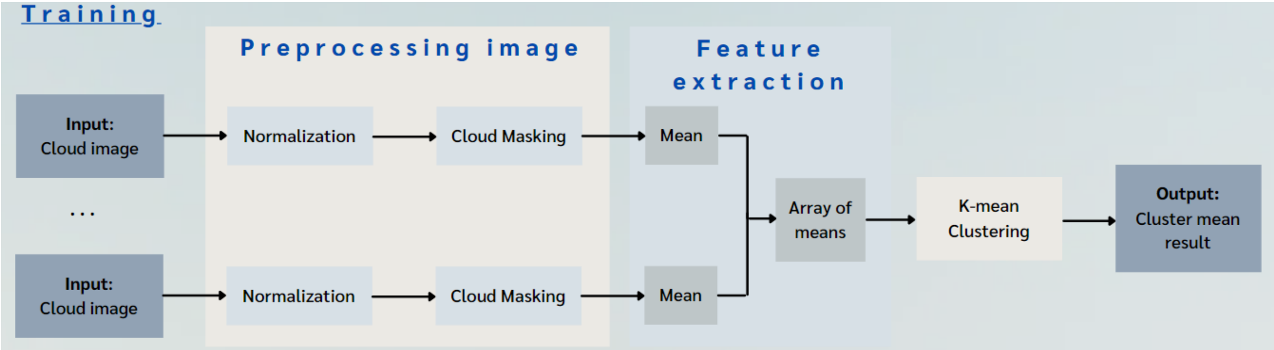
- Testing: HYTA Dataset, which includes 32 cloud images (many kinds of cloud)

Reference: https://github.com/Soumyabrata/HYTA/tree/master/images

3. Architecture (P.43)

We are using digital image processing techniques which process the images of the sky like normalization, cloud masking algorithm and k-mean algorithm.

3.1. Training architecture



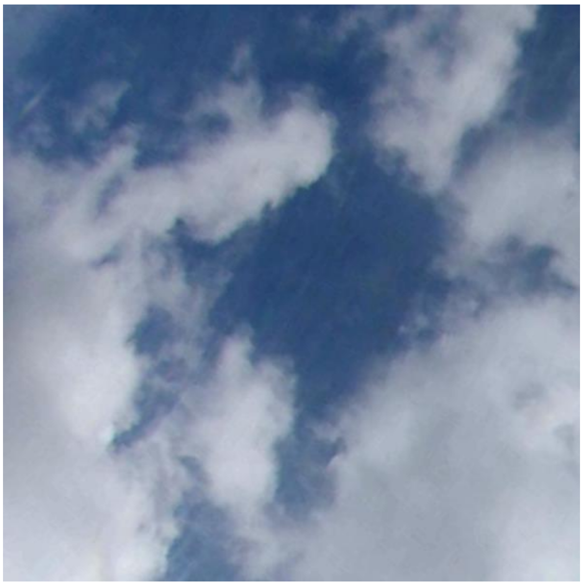
3.1.1. Normalization of Image

Step 1: Change the input image to grayscale image by adjusting the intensity range of the BGR pixels to all BGR values in that pixel.

<u>Code</u>: 0.2126*B+0.0722*G+0.7152*R

Step 2: Increase the intensity to get a clear distinction between the clouds and the sky. (Using arithmetic operations)

<u>Code</u>: f = (255/1)*(f/(255/1))**2



3.1.2. Cloud Masking Algorithm

Step 1: Threshold the image by using cv2.THRESH_TOZERO with threshold = mean intensity value of the image that will cut out all the information that has intensity lower than or equal to the threshold to keep only the cloud information of the image.

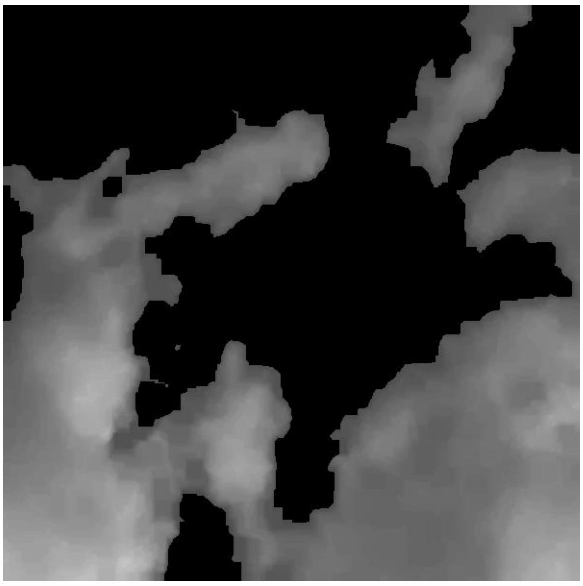
Code: _, n = cv2.threshold(f, mean_all_pixels, 255, cv2.THRESH_TOZERO)

$$\mathtt{dst}(x,y) = \begin{cases} \mathtt{src}(x,y) & \text{if } \mathtt{src}(x,y) > \mathtt{thresh} \\ 0 & \text{otherwise} \end{cases}$$

Step 2: Using morphology operation (closing) to fill some gaps between cloud areas and get little cloud edges that may be threshold out.

Code: kernel = np.ones((15, 15), np.uint8)

n = cv2.morphologyEx(n, cv2.MORPH_CLOSE, kernel)



3.1.3. Feature Extraction

Extract the feature of the cloud part by finding the mean value of the cloud area (where intensity value of that pixel is not equal to 0).

** divided by the number of pixels where intensity value not equal to 0 **

All of these processes (1-3) are done for all the images in the training and testing dataset.

3.1.4. K-mean clustering algorithm

Step 1: Initial cluster mean: [[min+1], [(min+Max)/2], [max-1]]

Step 2: For each training image, find the nearest cluster by using Euclidean distance.

Step 3: Add cluster size and label index for training image.

Step 4: Update mean of that cluster index.

Step 5: Do it until no more cluster index changes.

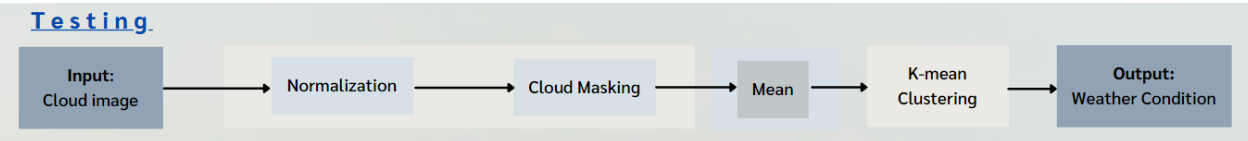
Step 6: Sort cluster mean.

Step 7: Label each cluster

(Cluster 0: Sunny, Cluster 1: Cloudy, Cluster 2: High chance of rain).

Cluster mean result = [[59.346], [112.251], [175.905]] (3 clusters)

3.2. Testing architecture



3.2.1. Do pre-processes and feature extraction like training dataset.

3.2.2. K-mean clustering algorithm

Compare cloud means and find the nearest cluster and return the result of weather conditions.

4. Result Analysis

If we use preprocessed images for training and testing, we will get 78.125% accuracy of weather prediction, which has higher accuracy than using normal images that have 62.5% accuracy.

5. Q&A

Q: ถ้าต้องการความแม่นยำที่เพิ่มมากขึ้นสามารถทำอย่างไรได้บ้าง

A: สามารถทำได้ด้วยการหารูป Dataset มาเพิ่ม / ใช้ Deep Learning มาช่วย โดยต้อง Label training และ testing dataset ด้วย / หาข้อมูลอื่นๆ เช่น อุณหภูมิ ความชื้นมาประกอบการ train model ด้วย

Q: ทำไมต้องแบ่ง K-mean clustering ออกเป็น 3 clusters

A: ตาม reference document แบ่งเป็นอากาศ 4 แบบ แต่เราแบ่งเป็น 3 clusters เพราะรวม clear sky กับ sunny เข้าเป็น cluster เดียวกัน

Q: ทำไมรูปที่มีดวงอาทิตย์เป็นองค์ประกอบถึงเกิดปัญหา

A: เพราะตอน normalize จะถูกตีว่าเป็นก้อนเมฆเพราะสีขาวเหมือนกัน ทำให้ cloud mean ของรูปนั้นผิดพลาดไป

Q: ข้อจำกัดของรูปที่เอามา test

A: รูปที่นำมา train และ test ต้องมีแค่ท้องฟ้ากับเมฆในรูปเท่านั้น ไม่ควรเห็นดวงอาทิตย์ชัดๆ

Q: หลักการกำหนดค่า mean ตั้งต้น

A: ก่อน train เราไม่รู้ว่า mean ของแต่ละ cluster เป็นเท่าไหร่ เลยต้องกำหนดที่ค่า [[min+1], [(min+max)/2], [max-1]]

Reference

Mrs. G. Gowri Pushpa , Hemaja Patoju , G. Sai Charan , M. Kali Charan, Pranita Jagtap from Department of Computer Science and Engineering ANITS, Vishakhapatnam, Andhra Pradesh, India, "Weather Forecasting Using Digital Image Processing"-JAC : A Journal Of Composition Theory (ISSN : 0731-6755).
