Contents:

Contents:

Abstract

To predict the conditions of the atmosphere for a given location Weather Forecasting is used. It is the application of science and technology. Weather forecast is more helpful for people as it predicts how the future weather is going to be and people may plan accordingly. Farmers will be the most beneficial one's as they may know the rainfall prediction and grow crops accordingly. The weather forecast can be done in many ways like using the previous data or analyzing the current clouds. We predict the weather using the status of the clouds. We used methodologies like Normalization, Clustering, and Cloud mask algorithm to predict the weather more accurately. Normalization is done using RGB values of each pixel. In many fields of research and in industrial and military applications Digital-image processing has become economical.

KEYWORDS :

Weather forecasting, normalization, clustering, cloud mask algorithm.

1. INTRODUCTION

1.1. Introduction :

Weather forecasting means predicting the weather and telling how the weather changes with change in time. Change in weather occurs due to movement or transfer of energy. Many meteorological patterns and features like anticyclones, depressions, thunderstorms, hurricanes and tornadoes occur due to the physical transfer of heat and moisture by convective processes. Clouds are formed by evaporation of water vapor. As the water cycle keeps on evolving the water content in the clouds increases which in turn leads to precipitation. This is how the convective process happens and also the change in weather. Many factors like temperature, rainfall, pressure, humidity, sunshine, wind and cloudiness are considered for predicting the weather. It is also possible to identify the different types of clouds associated with different patterns of weather. These patterns of weather help in predicting the weather forecast.

In the past, people used barometric pressure, current weather conditions, sky condition to predict whereas now there are many computer based models that consider the atmospheric factors to predict the weather. These methods are not accurate and the reason is due to the chaotic nature of the atmosphere as it keeps on changing. Even predicting weather for a longer period of time will not be accurate, that is why most of the current forecasting models predict weather only for a couple of days not more than

10. The accuracy gets reduced with increase in time.

Weather forecasting isn't a purely mechanical linear process, that standard practices and procedures will be directly applied. Forecaster’s job is predicated on theoretical background and lab work which needs several years of study but mainly day-to-day practice inside a weather forecasting service having a particular technical environment. The work of the forecasters has evolved significantly over the years to require advantage of both scientific and technological improvements. The skill of numerical models has improved such a lot that some centers are automating routine forecasts to permit forecasters to specialize in high impact weather or areas where they can add significant value. So it's dangerous to see a regular thanks to achieve weather forecasts.

1.1.1 Digital Image Processing :

The term digital image processing generally refers to processing of a two-dimensional picture by a digital computer. In a broader context, it implies digital processing of any two-dimensional data. A digital image is an array of real numbers represented by a finite number of bits. The principal advantage of Digital Image Processing methods is its versatility, repeatability and the preservation of original data precision.

- **Pixel:**

Pixel is the smallest element of an image. Each pixel corresponds to any one value. In an 8-bit gray scale image, the value of the pixel is between 0 and 255. The values of a pixel at any point correspond to the intensity of the light photons striking at that point. Each pixel stores a value proportional to the light intensity at that particular location.

- **Digital image:**

A digital image is nothing more than data— indicating variations of red, green, and blue at a particular location on a grid of pixels.

- **Gray level:**

The value of the pixel at any point denotes the intensity of image at that location, and that is also known as gray level.Generally to convert an image to grayscale, the equation that was used previously is :

Grayscale = (Red + Green + Blue / 3).

But as red has more wavelength we use the equation:

Grayscale = ((0.3 * R) + (0.59 * G) + (0.11 * B)).

Image Processing Techniques :

The basic definition of image processing refers to processing of digital image, i.e removing the noise and any kind of irregularities present in an image using the digital computer. The noise or irregularity may creep into the image either during its formation or during transformation etc. For mathematical analysis, an image may be defined as a two dimensional function f(x,y) where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the intensity values of f are all finite.
discrete quantities, we call the image a digital image. It is very important that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are called picture elements, image elements, pels, and pixels. Pixel is the most widely used term to denote the elements of a digital image.

The various Image Processing techniques are:

1) Image preprocessing

2) Image enhancement

3) Image segmentation

4) Feature extraction

5) Image classification

6) Image compression

7) Image restoration

8) Image acquisition

9) Image representation

10) Image fusion

11) Linear filtering .

1.1.2 Machine Learning :

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

Machine learning[10] is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is related field of study, focusing the exploratory data analysis using unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

MACHINE LEARNING TECHNIQUES:

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. For early tasks that humans assigned to computers, it was possible to create algorithms telling the machine how to execute all needed steps to solve the problem in hand. So on the computer's part, no learning was needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than have human programmers specify every needed step.

Early classifications for machine learning approaches sometimes divided them into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system. These were:

1) Supervised Learning 2) Unsupervised Learning 3) Semi Supervised Learning 4) Reinforcement Learning

- **SUPERVISED LEARNING:**

Some common supervised learning algorithms are:

1) Linear Regression **2) Logistic Regression**

3) Decision Tree **4) Random Forest**

5) KNN **6) SVM**

7) Naïve Bayes

- **UNSUPERVISED LEARNING :**

Some common unsupervised learning algorithms are:

1) K-means for clustering problems

2) Apriori algorithm for association rule learning problems

3) Principal Component Analysis

4) Singular Value Decomposition

5) Independent Component Analysis

- <u>**Linear Regression**</u> **:**

To understand the working functionality of this algorithm, imagine how you would arrange random logs of wood in increasing order of their weight. There is a catch; however – you cannot weigh each log. You have to guess its weight just by looking at the height and girth of the log (visual analysis) and arrange them using a combination of these visible parameters. This is what linear regression is like.

In this process, a relationship is established between independent and dependent variables by fitting them to a line. This line is known as the regression line and represented by a linear equation **Y= a *X + b.**

Where Y is the Dependent Variable

a is the Slope X is the Independent Variable

b is the Intercept

The coefficients a & b are derived by minimizing the sum of the squared difference of distance between data points and the regression line.

Real life applications of Linear Regression:

1) Risk Management in financial services or insurance domain

2) Predictive Analytic

3) Econometric

4) Epidemiology

5) Weather data analysis

6) Customer survey results analysis.

- **K-means CLUSTERING :**

K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. The 'means' in the K-means refers to averaging of the data; that is, finding the centroid.

K-means algorithm starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids. It halts creating and optimizing clusters when either the centroids have stabilized or a defined number of iterations have been achieved.

The K-means clustering algorithm:

1) Specify the number of clusters K.

2) Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.

3) Keep iterating until the centroids are stabilized.

4) Compute the sum of the squared distance between data points and all centroids.

5) Assign each data point to the closest cluster (centroid).

6) Compute the centroids for the clusters by taking the average of the data points that belong to each cluster.

1.2 Motivation For The Project :

Our motivation was to get the accurate result of the weather at that point of time. There were many applications and Google weather reports for predicting the weather but we wanted to forecast weather using image processing. All the applications need internet connectivity and Global Positioning System to say the weather at that place and at that time. Those are used to get the information or position of the user from the satellite and give the resultant weather from it. Every individual can get the weather just by taking a single digital photo of the cloud and sky at which they are standing. This idea motivated us to do this project.

1.3 Problem Statement :

To predict the weather at a particular area at any particular time with the help of the image of sky which should not contain any other objects other than the sky and clouds. Even in remote areas where people don't have access to the internet should also be able to get weather forecasts. To be able to solve the above problem, we are doing this project.

1.4. Organization of Thesis :

In this document, the next chapter consists of a literature survey. The literature survey tells about the research done to work on the project. All the details about the papers, websites on which the research work is done in order to work on the project is provided in the literature survey. In the methodology chapter , we discuss the various methodologies used in the project. In the Simulation & Results chapter, the details about experimental analysis are discussed. The experimental analysis includes sample code, result screenshots for a tested input image. In the conclusion chapter we give the conclusion about the project and also provide information if the project can be implemented further or not. In the final chapter we provide all the references for this project.

2. LITERATURE SURVEY

2.1. Weather Forecasting using satellite image processing & ANN :

The interpretation of satellite weather imagery has generally required the experience of a well-trained meteorologist. However, it is not always possible, or feasible to have an expert meteorologist on hand when such interpretation is desired. Therefore, the availability of an automated interpretation system would be quite desirable. Also, to take advantage of this available data in a reasonable and useful time increment, the system must be efficient and have low implementation cost. There are 3 main types of satellite images available - **Visible, Infrared** and **Water Vapor**. Visible images are obtained only during the day. They are used to determine the thickness of the clouds. Infrared images are obtained using special infrared sensors. The major advantage of this type is that it can be obtained even during night. It can be used to measure the temperature of cloud tops. Water Vapor images indicate the moisture content or humidity. The brighter areas tend to have high chances of rainfall.

The author uses satellite images for the detection of weather at that moment of time. This paper uses the strategy which uses ICA/Fast ICA Algorithm that is proposed by **Wang Yongqi and Du Huadong** i.e. Studies on **Cloud Detection of Atmosphere Remote Sensing using ICA algorithm**. In this algorithm three types of images are considered as input. For the separation of image normalization is done to the un-mixing matrix obtained from the input which is used to segment the clouds. Considering the image, in this paper 2 phases are done:

1)Image processing phase where image segmentation is done for the region of interest and using the normalization cloud cover over the required region is calculated in percentage. In this paper, **AABT** is designed to provide an accurate and fast segmentation. Those results are potentially accurate enough for cloud cover percentage calculation.

2) machine learning phase where the dataset is mapped with the calculated percentage obtained in image processing phase and the weather prediction is done by an artificial neural network. The unique combination of **NAR** and **NARX** neural network is used which produces positive results and accurate prediction to a very good extent.

For an automated weather satellite image interpretation system, one of the key steps is image segmentation. In this process, significant cloud features are extracted from the image and prepared for the next step in the process. AABT is designed to provide a fast and accurate method of image segmentation which is simple to implement as well. The segmentation results are provided quickly and with potentially enough accuracy to be integrated into a complete automated weather interpretation system or for cloud cover estimation. Furthermore, in case of the neural network model it can be successfully concluded from the above results that this unique combination of NAR and NARX neural network produce a positive result and the prediction is accurate to a good extent although there is always there lies a vast possibility of an error currently known as weather phenomenon[2] are such that the features required to be incorporated to create an extremely efficient model are very high, varied and in many cases incalculable. Although efforts in this area will always develop the scientific community as well as the world.

2.2. <u>Machine learning applied to weather forecasting:</u>

Weather forecasting is the task of predicting the state of the atmosphere at a future time and a specified location. Traditionally, this has been done through physical simulations in which the atmosphere is modeled as a fluid. The present state of the atmosphere is sampled, and the future state is computed by numerically solving the equations of fluid dynamics and thermodynamics. However, the system of ordinary differential equations that govern this physical model is unstable under perturbations, and uncertainties in the initial measurements of the atmospheric conditions and an incomplete understanding of complex atmospheric processes restrict the extent of accurate weather forecasting to a 10 day period, beyond which weather forecasts are significantly unreliableMachine learning, on the contrary, is relatively robust to perturbations and doesn’t require a complete understanding of the physical processes that govern the atmosphere. Therefore, machine learning may represent a viable alternative to physical models in weather forecasting.

Two machine learning algorithms were implemented: linear regression and a variation of functional regression. A corpus of historical weather data for Stanford, CA was obtained and used to train these algorithms.

The input to these algorithms was the weather data of the past two days, which include the maximum temperature, minimum temperature[3], mean humidity, mean atmospheric pressure, and weather classification for each day. The output was then the maximum and minimum temperatures for each of the next seven days.

In this paper, details of weather for the past 2 days are considered. Those details are considered as input and performing linear regression and variation of functional regression, output is obtained. The output is weather for the next 10 days. Generally the classification of weather gives 9 classes: clear, scattered clouds, partly cloudy, snow, thunderstorm, rain, overcast, fog, mostly cloudy. The dataset considered, classified all those into 3 classes: moderate cloudy, very cloudy, precipitation. The least mean square error for the linear regression and variation on functional regression is calculated and learning curves are drawn in this paper. Linear regression is low biased with a high variance model whereas functional is exactly opposite to it. Collection of more data can improve the linear regression model. Hence the author suggests to consider 4 to 5 days of data as input to the model.

Both linear regression and functional regression were outperformed by professional weather forecasting services, although the discrepancy in their performance decreased significantly for later days, indicating that over longer periods of time, our models may outperform professional ones. Linear regression proved to be a low bias, high variance model whereas functional regression proved to be a high bias, low variance model. Linear regression is inherently a high variance model as it is unstable to outliers, so one way to improve the linear regression model is by collection of more data.

Functional regression, how- 5 ever, was high bias, indicating that the choice of model was poor, and that its predictions cannot be improved by further collection of data. This bias could be due to the design choice to forecast weather based upon the weather of the past two days, which may be too short to capture trends in weather that functional regression requires. If the forecast were instead based upon the weather of the past four or five days, the bias of the functional regression model could likely be reduced. However, this would require much more computation time along with retraining of the weight vector w, so this will be deferred to future work.

2.3 Analysis on various techniques for weather forecasting:

- **Support Vector Machines :**

  To predict the maximum temperature of a required location Support Vector Regression (SVR) is used. It performs better than MLP which is trained with back propagation algorithms as it minimizes the upper bound on generalization error. By selecting proper parameters it can replace neural networks based models for applications of weather prediction.
- **Time Series Analysis for Weather Forecasting :**

  Data groups and data variables in the specified time are captured by Time Series Analysis. By comparing actual and predicted values of temperature, the forecasting reliability was evaluated. The results show that an important tool for temperature forecasting is the network.
- **Prediction of Weather by using Back Propagation Algorithm :** Wind, humidity, rainfall and temperature are the parameters recorded using sensors. Using these sensors weather forecasting and processing information is transferred. It classifies, compares and predicts the change in other weather parameters by changing any one parameter value that those sensors recorded. A 3 layered neural network trained with the existing dataset to develop a relation among the parameters of weather that are non linear.
- **Fuzzy Logic Based Rainfall Prediction model:**

  Two components are made in a developed fuzzy logic model where one is knowledge based and the other is fuzzy reasoning or decision making. Using fuzzification and defuzzification operations outputs are predicted compared with actual rainfall data. A fuzzy model that is well developed is capable of handling the data that is scattered and shows flexibility in modeling weak input and output variable relationships.

2.4 Weather forecasting using data mining research based on cloud computing :

Weather Prediction is the application of science and technology to predict atmospheric conditions ahead of time for a particular region. Prediction is one of the basic goals of Data Mining. Data Mining is to dig out knowledge and rules, which are hidden and unknown. Users may be interested in or have potential value for decision-making from the large amounts of data. Such potential knowledge and rules can reveal the laws between the data. There are many kinds of technical methods of data mining, which mainly include: association rule mining algorithm, decision tree classification algorithm, clustering algorithm and time series mining algorithm, etc. How to store, manage and use these massive meteorological data, discover and understand the law and knowledge of the data, to contribute to weather forecasting completely and effectively has attracted more and more Data Mining researcher's attention. This article constructs the Weather Forecasting platform, using data mining for meteorological forecasts and the forecast results are analyzed.

Cloud computing has improved the efficiency of data storage, delivery, and dissemination across multiple platforms and applications, allowing easier collaboration and data sharing, including data processing and distribution systems that disseminate key weather forecasting, severe weather warning, and climate information. Data mining techniques and forecasting applications are very much needed in the cloud computing paradigm. In this study, data mining in Cloud Computing allows weather forecasting and data storage, with assurance of efficient, reliable and secure services for their users. The implementation of data mining techniques through Cloud computing will allow the users to retrieve meaningful information from virtually integrated data warehouses that will reduce the costs of infrastructure and storage.

A modern method is developed which is service oriented architecture for the weather information systems that forecasts weather using data mining techniques. The method uses Artificial Neural Network and Decision tree Algorithms and meteorological data collected in specific time. It presents the best results for generating classification rules for the mean weather variables. The model predicts temperature, rainfall and wind speed. Cloud computing reduces the cost of infrastructure and storage as it ensures secure, reliable and efficient services for the user.

The implementation of data mining approach to solve the wind forecasting problems for wind farm production, in particular, for predicting wind speed. The data mining prediction algorithm-ARIMA time series prediction algorithm is also integrated into the system. The platform has the ability of mass storage of meteorological data, efficient query and analysis, weather forecasting and other functions. In this study we also adapted the method of Artificial Neural Networks. It can detect the relationships between the input variables and generate outputs based on the observed patterns inherent in the data without any need for programming or developing complex equations to model these relationships. An artificial neural network (multi-layer perceptron) was applied and several simulations have been conducted for comparison purposes. ANN's can detect the relationships between weather parameters and use these for future prediction. Weather conditions are important to climatic change studies because the variation in weather conditions in terms of temperature and wind speed can be studied using these data mining techniques. ANNs are implemented, in order to compare their effectiveness in changing the network topology and the training mode. The results obtained from real data are based on a time series of meteorological data provided by the Dalian Meteorological Bureau. The test cases pointed out that the proposed approach gives a very interesting performance of the implemented network and shows good performance in terms of MSE. For future perspective, there is still significant potential for improvement in weather forecasting by using ANN model, through introducing climate change and global warming variables, in order to forecast more realistic weather parameters.

2.5 Cloud image analysis and classification :

Clouds are classified by their shape, temperature, color, density, spectral clustering analysis, training of rule based systems or neural networks and image processing techniques. Classification of clouds are highly time dependent because temperature, type of cloud dependence changes in different latitudes through different seasons. Different types of cloud are present out of which three basic cloud forms are the Cirrus, Stratus, and Cumulus. These forms are further refined into 10 other types based on their height and texture.

1.Cirrus: Thin, white and feathery appearance and mostly white patches or narrow bands.

2.Cirrocumulus: Thin white bands or ripples, sheet, or layered of clouds without shading.

3.Cirrostratus: High, milky white like appearance. They are transparent, whitish veil clouds with a fibrous (hair-like) or smooth appearance.

4. Altocumulus: Bumpy rounded masses, cotton ball appearance, white and/or gray patch sheet or layered clouds.

5. Altostratus: Transparent blue/gray clouds sheets or fibrous clouds that totally or partially cover the sky.

6. Nimbostratus: They are continuous rain clouds also known as storm clouds.

7. Stratocumulus: Gray or whitish layer with sheet, or layered clouds which almost always are dark.

8. Stratus: Cover large portion of sky, thin, sheet-like, gray and thick.

9. Cumulus: Cauliflower-like appearance with bulging upper parts.

10. Cumulonimbus: The thunderstorm cloud, heavy and dense cloud in the form of a mountain or huge tower.

2.6. Existing System :

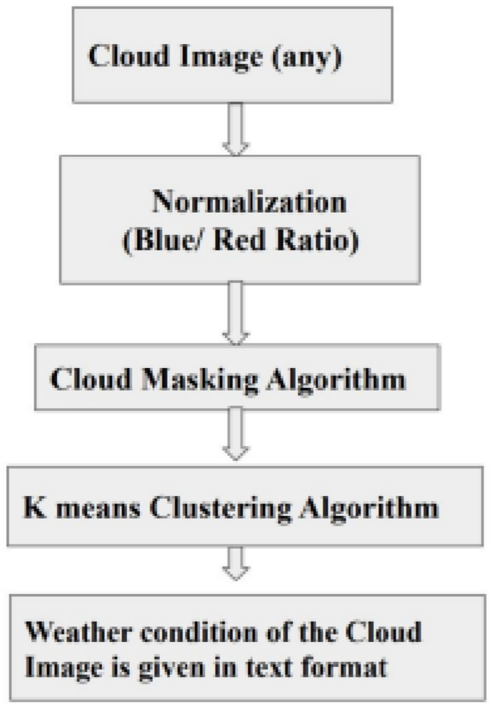
The existing system for weather forecasting mainly uses satellite images to predict the weather. Most of the websites which give weather forecasts with the help of GPS as well as satellite images data to predict the weather. Very rarely we find applications or websites predicting weather using digital image processing. These applications or websites use the satellite images in which we get the data about the cloud cover over any particular region and by using this data they predict the weather. They track the movement of the clouds for a particular time period and use that to predict weather for the future. This is the existing system for weather forecasting.

3. METHODOLOGY

3.1. PROPOSED SYSTEM :

Weather forecasting can also be done by using **satellite images** but acquiring the satellite images is more difficult and would even cost high. Even predicting using the satellite images needs more technology. So, we are using digital image processing techniques which process the images of the sky like **Normalization, Cloud Masking algorithm** and **k-Means algorithm.**

Architecture :



Every system should be divided into modules for better understanding and execution. Dividing into modules helps the programmer and client to work and use the system efficiently, respectively. If any system is not divided into modules and worked as a whole, then there comes numerous errors. Even we find difficulty in correcting those errors. It is must and should divide the total project into modules and work on each and every module independently to get effective results. Our total system is divided into three modules namely:

1) Normalization of Image, 2) Cloud masking algorithm, 3) K-means clustering

3.2. Normalization of Image :

- **Algorithm :**

Step 1: Pixel values for each and every pixel are considered. Pixel value consists of red, blue and green color's values. These values are extracted from the image with the help of predefined libraries in python.

Step 2: Now with the help of these pixels, we must change the intensity range of the pixels to [0,1] and increase the intensity to get a clear distinction between the clouds and the sky. Hence the digital picture is normalized.

Step 3: The input image can be of any digital image with the extension .jpg, .jpeg, .png.

Step 4: The output of this module would be a normalized image of the given digital image which seems likely to be a black and white or gray scale image.

We used different formulas to get the image normalized using the red, blue and green values of each pixel. And the threshold value is generated by taking the mean of all the pixel values. The input image can be of any digital image with the extension .jpg, .jpeg, .png. And the size of the image must vary between 20kb to 20mb. The output of this module would be a normalized image of the given digital image which seems likely to be a black and white or gray scale image.

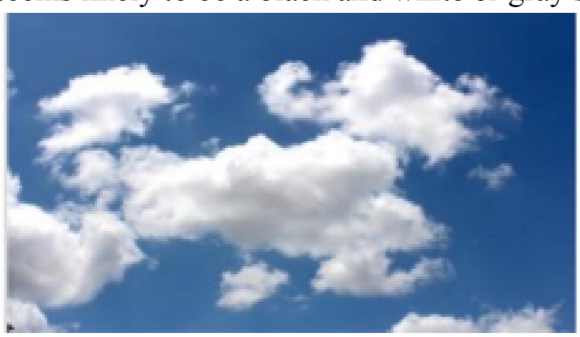


Fig 1: Sample Original Image of Cloud

Fig 2. Image After Performing Normalization

3.3. Cloud Masking Algorithm :

Considering the threshold value generated in the normalization module, we differentiate the cloud from the sky of the input image so that we can get the area of the cloud upon which we will perform further operations like feature extraction.

The data set is considered and normalization is done for each and every image in it. Then after getting the clouds separated from the image, the mean point is derived from each cloud. Based on those mean points, clouds are separated. Comparing the input image threshold value and mean value of the cloud, it would be pushed into that category of cloud for which its values coincide. The output of this module would be like in Fig 3.

- **Algorithm :**

Step 1: After normalization, a mean value is generated by adding all the pixel values and by dividing it by the total no of pixels. With the help of this mean value we differentiate the clouds from the input image.

Step 2: Now extract the feature of the cloud part by again finding the mean value of the cloud area which will be used as a feature in the next process.

Step 3: This process is done for all the images in the dataset so that we get features of all images which will be used to cluster the images into groups.

Fig. 3- Image obtained after performing Cloud Mask Algorithm

3.4. K means Clustering :

We considered clustering because for classification there would be less no of classes. But we considered ten types and hence we considered clustering rather than classification. Here we considered the clusters of the clouds as we would divide the image based on the cloud mean point.

In this the clouds are divided into 10 clusters. The clusters are:

1. **Cirrostratus**
2. **Cirrus**
3. **Cirrocumulus**
4. **Altocumulus**
5. **Altostratus**
6. **Stratus**
7. **Stratocumulus**
8. **Nimbostratus**
9. **Cumulonimbus**
10. **Cumulus**

Among these the **cirrostratus, altostratus, cirrocumulus, cirrus** denote sunny days. **Nimbostratus and Cumulonimbus** denote rainy days. And **cumulus, stratus, altocumulus** and **stratocumulus** denote cloudy days.

All the classification can be done depending on the mean threshold value. From the dataset after applying normalization and cloud masking algorithm we can get a threshold value for each and every cloud cluster. Based on that value i.e. that threshold value the input image is classified into one of the ten clusters. Then based on that cluster type the type of cloud can be predicted. After that based on the type of the cloud weather can be forecasted as per the above argument.

- **Algorithm :**

1.First we initialize k points, called means, randomly.

2. We categorize each item to its closest mean and we update the mean's coordinates, which are the averages of the items categorized in that mean so far.

3. We repeat the process for a given number of iterations and at the end, we have our clusters.

4.EXPERIMENTAL ANALYSIS AND RESULTS

4.1. SYSTEM CONFIGURATION:

4.1.1. Software Requirements:

1.A Python IDE. We have used **Google Colaboratory.**
2.Many External Library such as **Numpy, OpenCv, Panda library.**

- **Google Colaboratory:**

Google Colab, or 'Colaboratory', allows us to write and execute Python in our browser, with

1. Zero configuration required
2. Free access to GPUs
3. Easy sharing

- **OpenCv :**

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. OpenCV supports some models from deep learning frameworks like TensorFlow, Torch, PyTorch (after converting to an ONNX model) and Caffe according to a defined list of supported layers. It promotes OpenVisionCapsules. which is a portable format, compatible with all other formats.

- **Numpy :**

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy is open-source software and has many contributors.

- **Panda :**

Pandas is a Python library for data analysis. Started by Wes McKinney in 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas has grown into one of the most popular Python libraries. It has an extremely active community of contributors.

4.1.2. Hardware Requirements:

1) Processor: 64 bit, quad-core, 2.5 GHz minimum per core

2) Ram: 4 GB or more

3) Hard disk: 20 GB of available space or more.

4) Display: Dual XGA (1024 x 768) or higher resolution monitors

4.2. Experimental Analysis :

- **Image Normalization :**

Implementation of the total module is completed. The input image can be of any digital image with the extension .jpg, .jpeg, .png. And the size of the image must vary between 20kb to 20mb. The output of this 59 module would be a normalized image of the given digital image which seems likely to be a black and white or gray scale image.

Given Input :

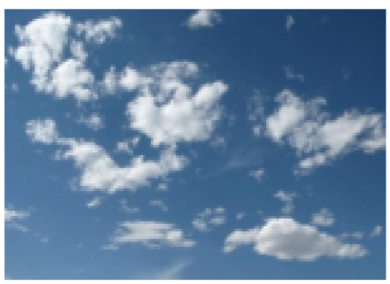


Fig 1 : Input Image 1

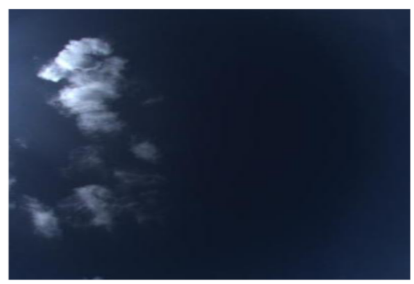


Fig 2: Input Image 2

Fig 3: Input Image 3

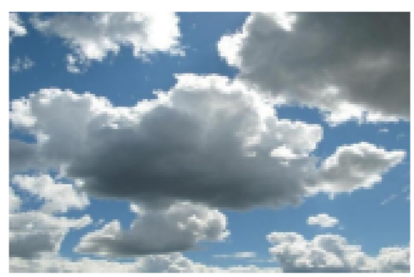


Fig 4: Input Image 4

- **Observed Output Images After Applying Normalization:**

Fig 5: Normalized Output Image 1

Fig 6: Normalized Output Image 2

Fig 7: Normalized Output Image 3

Fig 8: Normalized Output Image 4

- **Cloud Masking Algorithm :**

When coming to the status of implementation, code has been developed up to a certain extent where the cloud detection is performed. The feature extraction of the cloud still needs to be done and also the training of the dataset is yet to be started.

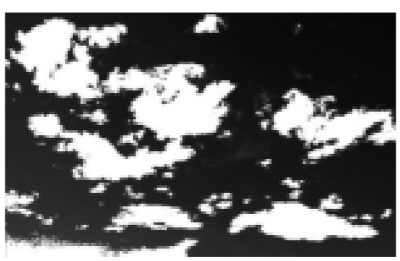


Fig 9: Output Result 1

Fig 10: Output Result 2

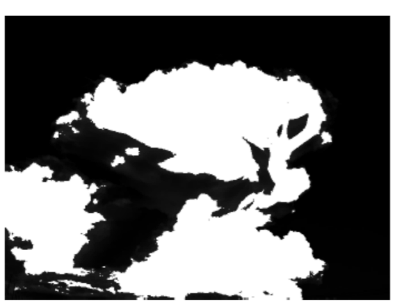


Fig 11: Output Result 3

Fig 12: Output Result 4

- **K-Means Clustering :**

K-means clustering is done on the dataset to cluster the images in the dataset into 4 clusters. This process is done by obtaining the means of the images which is the main feature considered for clustering. From the list containing means we consider ‘k’ random means as the initial mean[6] points. From these points we calculate distance between the other means with the help of Euclidean distance formula:

Sqrt (S += math.pow(x-y, 2))

After finding the distances, we update the randomly selected means based on the distances. This is an iteration process which is repeated until there is no change in the randomly selected means. Once the process is complete we get the means of the clusters which are considered as the centroid of the cluster. Based on these means we can find to which cluster the given input image belongs.

EXPERIMENTAL ANALYSIS/ TESTING:

Dataset:

The dataset we considered is named “HYTA”. It consists of various images of all types of clouds. We considered 4 clusters for all the types of clouds namely: clear sky, sunny, cloudy and sunny, rainy. For each type of cluster this HYTA dataset consists of nearly 8 to 10 images. Every image consists of only a plain sky with respective clouds and no other objects like buildings, trees and poles. In some images the sun might appear along with the sky and clouds. Along with this standard dataset we considered 4 different datasets. Every data set consists of more than 10 photos for each type of cloud. These datasets are considered to compare the outputs obtained and check the accuracy of the model developed. This comparison will be useful for the future development of the model.

4.3 Final Result :

After doing the entire Process, We get the final result i.e the weather condition on the basis of the 4 Cloud Input Images taken.

1. In case of Input Image 1, the weather condition as output is : **SUNNY.**

2. In case of Input Image 2, the weather condition as output is : **CLEAR SKY**

3. In case of Input Image 3, the weather condition as output is : **CLOUDY AND SUNNY**

4. In case of Input Image 4, the weather condition as output is : **CLOUDY WITH CHANCES OF RAIN.**

CONCLUSION

Generally, all the other weather forecasting applications and sources would give the weather report of that particular area. Using the GPS, location would be tracked and using satellite information, weather conditions would be given at that place. The types of clouds the author considered, can deliberately give accurate conditions of the weather. For now, the model can give the weather condition at that point of time.

Future Work

To get weather forecast for the next few days, we can modify our system by using different algorithms and use that as an extension to our current project.

REFERENCES

1. https://www.mathworks.com/matlabcentral/answers/422493-how-to-donormalized blue-red-ratio-operation-of-an-image

2. Calbo, J., and J. Sabburg, 2008: Feature extraction from whole-sky ground-based images for cloud-type recognition. J. Atmos. Oceanic Technol., 25, 3–14

3. https://www.geeksforgeeks.org/k-means-clustering-introduction/

4.https://www.science-emergence.com/Articles/How-to-create-a-scatter-plot-with-several-colors-in-matplotlib-/

5. https://medium.com/sentinel-hub/improving-cloud-detection-with-machinelearning-c09dc5d7cf1 3

6. https://github.com/Soumyabrata/HYTA

7. https://drive.google.com/drive/folders/1v_PzsbNhk5a7lsa8M6xi-iv29UdYfb96?usp=sharing

7. Minakshi Gogoi and Gitanjali Devi (Oct-Dec 2015), Cloud image analysis for rainfall prediction, Advanced Research in EEE.

PROJECT CODE :

```
from google.colab import drive

drive.mount("/content/gdrive")

# for Mounting the dataset from the google drive


import cv2

import numpy as np

import os

import pandas as pd

import random as rd,math

import sys



path='/content/gdrive/MyDrive/Cloud Image Dataset'

files=[]

meansl=[]

m=[]

M=[]

c=0

for r,d,f in os.walk(path):

  for file in f:

    if('.jpeg' in file or '.jpg' in file):

      files.append(os.path.join(r,file))

      c+=1

#print(files)

for input_file in files:

  f=cv2.imread(input_file)

  h, w, bpp = np.shape(f)

  for py in range(0, h):

    for px in range(0, w):

      x = float(f[py][px][0])

      y = float(f[py][px][1])
```

```
    z = float(f[py][px][2])
    f[py][px] = x * 0.2126 + y * 0.0722 + z * 0.7152
  f = (255 / 1) * (f / (255 / 1)) ** 2
  h, w, bpp = np.shape(f)
  n = np.zeros([h, w, bpp], dtype=np.uint8)
  c = 0
  sum = 0
  for py in range(0, h):
   for px in range(0, w):
    sum += f[py][px][0]
    c += 1
  for py in range(0, h):
   for px in range(0, w):
    if f[py][px][0] > (sum / c):
     n[py][px] = f[py][px]
     f[py][px] = 255
  sum1 = 0
  z = 0
  for py in range(0, h):
   for px in range(0, w):
    if n[py][px][0] != 0:
     sum1 += n[py][px][0]
     z += 1
  mean = sum1 / z
  dict={}
  dict1={}
  dict[input_file]=mean
  dict1=mean
  '''print(dict1)'''
  meansl.append(dict1)
'''print(meansl)'''
M.append(max(meansl,default=0))
m.append(min(meansl,default=0))
```

```
def InitializeMeans(meansl, k, m, M):
 f=1;#no. of features
 means=[[0 for i in range(f)] for j in range(k)];
 for item in means:
  for j in range(len(item)):
   item[j]= rd.uniform(m[j]+1,M[j]-1);
 return means;

def EuclideanDistance(x,y):
 S=0;
 for i in range(1):
  S += math.pow(x-y, 2);
 return math.sqrt(S);

def UpdateMean(n, mean, item):
 for i in range(len(mean)):
  m2 = mean[i];
  m2 = (m2 * (n - 1) + item) / float(n);
  mean[i] = round(m2, 3);
 return mean;

def CalculateMeans(k, items, maxIterations=100000):
 cMin=m;
 cMax=M;
 # Initialize means at random points
 means = InitializeMeans(meansl, k, cMin, cMax);

 # Initialize clusters, the array to hold
 # the number of items in a class
 clusterSizes = [0 for i in range(len(means))];
 # An array to hold the cluster an item is in
 belongsTo = [0 for i in range(len(meansl))];

 # Calculate means
```

```
  for e in range(maxIterations):
   # If no change of cluster occurs, halt
   noChange = True;
   for i in range(len(meansl)):
    item = meansl[i];

    # Classify item into a cluster and update the
    # corresponding means.
    index = Classify(means, item);

    clusterSizes[index] += 1;
    cSize = clusterSizes[index];
    means[index] = UpdateMean(cSize, means[index], item);

    # Item changed cluster
    if (index != belongsTo[i]):
     noChange = False;

    belongsTo[i] = index;

    # Nothing changed, return
   if (noChange):
    break;
   return means;
 def Classify(means, item):
  minimum = sys.maxsize;
  index = -1;

  for i in range(len(means)):

   dis = EuclideanDistance(item, means[i]);

   if (dis < minimum):
    minimum = dis;
```

```
      index = i;

  return index;

def FindClusters(means, meansl):
  clusters = [[] for i in range(len(means))]; # Initialize clusters
  for item in meansl:
    index = Classify(means, item);

    clusters[index].append(item);

  return clusters;

means=CalculateMeans(4,meansl)
means.sort()
print(means)
g=FindClusters(means,meansl)
print(g)

m = cv2.imread('/content/gdrive/MyDrive/Cloud Image
                                              Dataset/Input14.jpg'
                                              )
print('For The Selected IMAGE :')

from google.colab.patches import cv2_imshow
cv2_imshow(m)

h,w,bpp = np.shape(m)
red=[]
blue=[]
green=[]
for py in range(0,h):
  for px in range(0,w):
    red.append(m[py][px][0])
    blue.append(m[py][px][1])
```

```
    green.append(m[py][px][2])
red_max=max(red)
blue_max=max(blue)
green_max=max(green)
red_min=min(red)
blue_min=min(blue)
green_min=min(green)
for py in range(0,h):
  for px in range(0,w):
    x=float(m[py][px][0])
    y=float(m[py][px][1])
    z=float(m[py][px][2])

    m[py][px]=x*0.2126+y*0.0722+z*0.7152

m = (255 / 1) * (m / (255 / 1)) ** 2

print('Selected IMAGE after Normalization :')
cv2_imshow(m)
h,w,bpp = np.shape(m)
red=[]
blue=[]
green=[]
for py in range(0,h):
  for px in range(0,w):
    red.append(m[py][px][0])
    blue.append(m[py][px][1])
    green.append(m[py][px][2])
red_max=max(red)
blue_max=max(blue)
green_max=max(green)
red_min=min(red)
blue_min=min(blue)
green_min=min(green)
```

```
for py in range(0,h):
  for px in range(0,w):
    x=float(m[py][px][0])
    y=float(m[py][px][1])
    z=float(m[py][px][2])

    m[py][px]=x*0.2126+y*0.0722+z*0.7152

m = (255 / 1) * (m / (255 / 1)) ** 2
print('Selected IMAGE after Cloud Masking Algorithm :')
h,w,bpp = np.shape(m)
n=np.zeros([h,w,bpp], dtype=np.uint8)
c=0
sum=0
for py in range(0,h):
  for px in range(0,w):
    sum+=m[py][px][0]
    c+=1
for py in range(0,h):
  for px in range(0,w):
    if m[py][px][0]>(sum/c):
      n[py][px]=m[py][px]
      m[py][px]=255
sum1=0
z=0
for py in range(0,h):
  for px in range(0,w):
    if n[py][px][0]!=0:
      sum1+=n[py][px][0]
      z+=1
mean=sum1/z
cv2_imshow(m)
print(mean)
for i in range(len(means)):
```

```
 if mean>means[i]:
  v=i;
 if v!=len(means)-1:
  o1=means[v+1]-mean;
  o2=mean-means[v];
  if o1<o2:
   v=v+1;
  '''print(v)'''
 else:
  v=len(means)-1;
'''v=3'''
print('Current Weather Condition is: ')
if v==0:
 print('CLEAR SKY')
elif v==1:
 print('SUNNY')
elif v==2:
 print('CLOUDY AND SUNNY')
elif v==3:
 print('CLOUDY WITH CHANCES OF RAIN')
```