PROJECT REPORT

FOR

EE592P "SELECTED TOPICS ON IoT"

ON

SHORT-TERM WEATHER FORECASTING USING COMBINATION OF GROUND BASED SKY IMAGERY WITH TEMPERATURE AND HUMIDITY SENSOR VALUE

SUBMITTED BY

JOE JOHNSON (D17024)

PhD SCHOLAR, SCEE, IIT MANDI

UNDER THE GUIDANCE OF

SRIKANT SRINIVASAN (ASSISTANT PROFESSOR)

AND

SIDDHATHA SARMA (ASSISTANT PROFESSOR)

SCEE, IIT MANDI

CONTENTS

1. INTRODUCTION

Getting a short-term weather forecast could be useful in a place like Himachal Pradesh, where the weather is dynamic. This could help in power grid optimization, smart irrigation and cattle management. As there is recent trend in increasing use of renewable energies and source of most energies are directly dependent on local weather. Even though satellite weather forecasting is more accurate and forecast period is from few seconds to four hours, its cost does not justify the benefits. Here the project is implemented using common IoT devices and sensors for the project. Three Picameras facing the sky with raspberry Pi to cover three directions of sky has been used and data readings from temperature and pressure sensors for optimized predictions for time of the day. The cloud, shadow on mountain, building and other objects could be detected with aid colour intensity histogram of predefined image patches using open source library OpenCV and using PCA to reduce the dimension of feature, and use support vector classifier to predict rain by sunny or dark cloud sky detection. Stages of project include setting up of cameras and sensors, then taking training data, fit a model and predict test data real-time. Later the model could used to forecast weather with given live images and sensor data. The model is expected to give short-term weather forecast of few seconds to minutes.

2. SETUP

The prototype of setup include Raspberry pi3 which commonly used SoC for IoT, PiCamera V2.1, Sense HAT module, Arduino Uno (Microcontroller 328p) and DS18B20 temperature sensor. The complete setup placed towards west direction of MANAS lab has been shown in Fig 1. As there is no GPIO terminal available for Raspberry Pi 3 module after interfacing sense HAT module, DS18B20 sensor is interface through Arduino Uno to Pi. This setup is replicate at north direction and east direction placed at IoT lab.

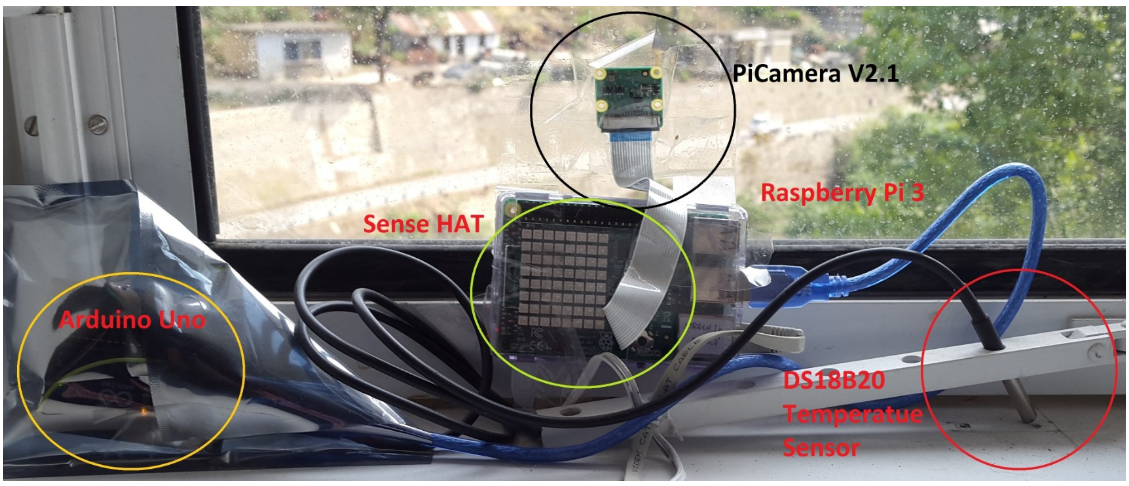

Fig 1. Hardware setup for weather forecast facing west direction of MANAS lab.

The figure 2 and figure 3 shows the set towards east and north direction placed at IoT lab respectively. The setup does not require any special weather proof casing as setup is placed behind glass window panel. For all three setup internet connectivity is provided by Ethernet cable.

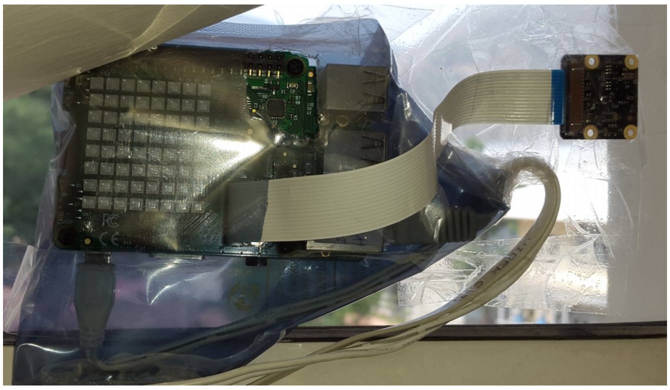

Fig 2. Hardware setup for weather forecast facing east direction of IoT lab.

Fig 3. Hardware setup for weather forecast facing north direction of IoT lab.

3. FEATURE EXTRACTION

Here the image taken are of resolution 480X360 and made into 400 patches using 24 pixels window for column and 18 pixels window for row. Using OpenCV library, every patch are converted to R,G,B separate images and histogram of range 0 to 255 was calculated. Again these 256 histogram values are converted to 32 bins of 8 histogram values each. Each image will provide 400X32 feature matrix. These 32 feature dimensions are reduced to one principle component by applying PCA. The reduced 400X1 column vector are arranged to row vector for each channel and for each image. The training set are made using ten image feature for each class, here two class i.e., sunny and cloudy sky are considered. PCA is generated using SciKit Learn library.

4. CLASSIFIER

For classifying the test samples real-time, support vector machine (SVM) has been used. The advantage of choosing SVM is that the classifier gives better result with less training samples as it is free from curse of dimensionality. For implementing the classifier, SciKit learn library has been used. The “model = svm.SVC(kernel='rbf', C=1, gamma=1)” command is for choosing radial basis function kernel for the classifier, “model.fit(classes, type_label)” command to fit the training features with class label and “model.predict([[x_test, y_test]])” command to predict the class for given input data. The classifier gave 100 percent accuracy for predefined test dataset.

5. IMPLEMENTATION

The training dataset has been made with all three setup facing different directions, ten logical images for each class has been selected for training and testing. The reduced dimension feature dataset has been created and saved as Tr_all.csv and Ts_all.csv file. The sample image for sunny training dataset is shown in Fig 4 and cloudy image for the same is shown in Fig 5.

Fig 4. Sunny dataset image for west direction

Fig 5. Cloudy dataset image for west direction

The process of capturing image to producing features take 90 seconds using Raspberry Pi3 board, hence computationally expensive. The image captured by north facing camera on a sunny day is shown in Fig 6 and during cloudy day is shown in Fig 7.

Fig 6. Sunny dataset image for north direction

Fig 7. Cloudy dataset image for north direction

The image captured for east direction is using NoIR Pi camera, the image is shown in Fig 8.

Fig 8. Sunny dataset image for east direction

For the image data the observation are that, the cloudy instance are observable by dark sky and shadow on hills with thick fog. The intensity of the histogram changes drastically.

The data from various sensor are stored in text form for keep track of changes in different parameter to and trends in past. The parameter stored are DS18B20 temperature, sense HAT temperature, pressure and humidity for every 5minutes (300 sec). For better visual understanding and sharing data globally, the ThingSpeak tool is used. After each process, the data is sent to three ThingSpeak channel each for each directions. Figure 9 shows the channel status for west side data. Field 1 shows the change in DS18B20 temperature sensor reading w.r.t time and field 2 gives change in HAT temperature sensor w.r.t time.

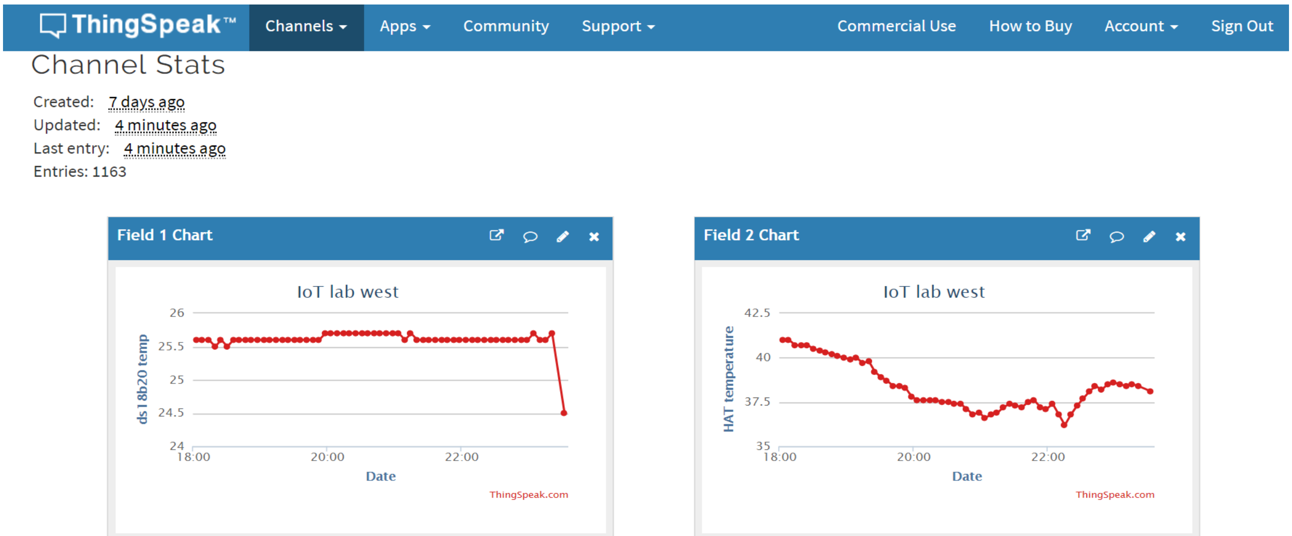

Fig 9. Data plotted in ThingSpeak online tool for temperature sensor w.r.t time for west direction

The Fig 10 shows the change in pressure w.r.t time which is assigned to field3 and the change in humidity w.r.t time assigned to field 4.

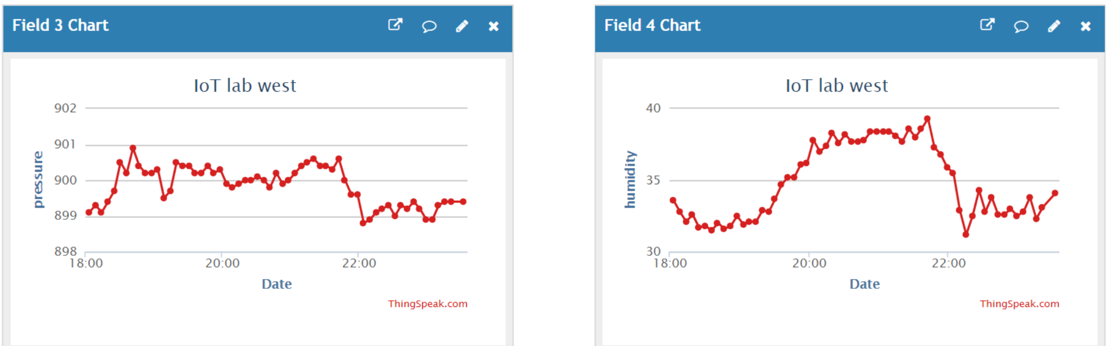

Fig 10. Data plotted in ThingSpeak online tool for pressure and humidity sensor w.r.t time for west direction

From the above four plots it has been found that with increased chances of rain, the temperature reduces and humidity tends to increase, there is no inferable change w.r.t weather for pressure measurement. SVM classifier is also built on temperature and humidity parameter. Figure 11 gives the channel location and updates the predicted status of weather after each iteration. The status updates the sunny or cloudy state of local weather. As there is no separate class for night the classifier predicts the night as cloudy due to similarity in conditions.

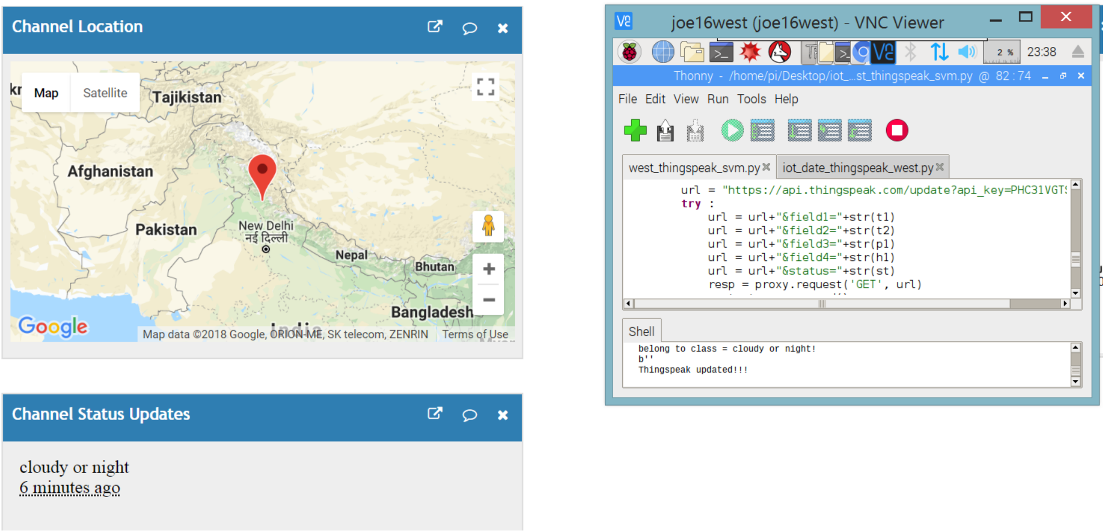

Fig 11. Data is updated in ThingSpeak online tool for predicted state of local weather with location for west direction

6. CONCLUSIONS

The system was successfully able to classify the different state of weather locally with good accuracy, for every iteration the external temperature, internal temperature, pressure and humidity are plotted real time and saved. The reduction in dimensionality does not effect the classifier’s ability. It was observed that even with less number of sample values the classifier has predicted properly. On an average the system predicts the weather five minutes ahead of occurrence.

7. FUTURE SCOPE

More states of weather, for example wind could be modelled by observing the pixel motion by using opticalflow or other methods.

8. REFERENCES

1. https://picamera.readthedocs.io/en/release-1.10/recipes1.html

2. https://www.raspberrypi.org/blog/sense-hat-projects/

3. https://create.arduino.cc/projecthub/TheGadgetBoy/ds18b20-digital-temperature-sensor-and-arduino-9cc806

4. http://www.instructables.com/id/ThingSpeak-Temperature-Monitor-with-Raspberry-Pi/

5. http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html