Classification of weather using CNN and deep learning

Byreddy Vishnu, Pranjal Pandey, Suhas Cholleti

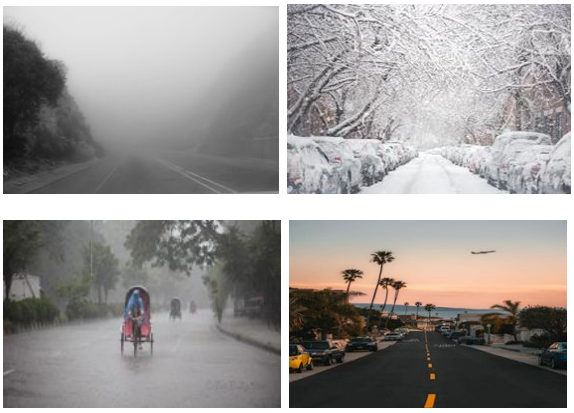
***Abstract*—Weather plays a huge part in day to day life, from destroying crops to affecting people's commute it affects every individual by one way or another. Generally, in traditional methods we always make use of expensive sensors and human involvement to detect the weather. It often ends up giving results with low accuracy. There's also a constraint on the availability of human resources and expensive instruments .Hence to save the human efforts and instruments the new approach of classifying the approach has been proposed. This approach involves use of computer vision and machine learning to classify the weather condition based on the image. We can make use of surveillance cameras which are readily available to get the image of the outdoors and detect the weather based on the Image and time.**

I. Introduction

With the increase in automation and self driving cars right around the corner, it is important that a computer understands the weather it is currently in. Depending on whether it is raining or snowing or clear skies the car has to decide at what speed it needs to be. Determining the weather from a single image is a simple task for an average human but designing a classifier that does this from a single image is a tall order. In recent years, computer vision is used to extract and understand the features in an image.

This paper is concerned to predict the present weather conditions from the image given to it. We also want to try out multiple machine learning techniques and compare them so that we can choose the best for our purposes. One such technique is Constitutional Neural Networks(CNN). CNN uses perceptrons for supervised learning, to be able to analyze and classify images.

In this project, a weather classification method has been proposed which involves automatically elicitation of information by using Computer vision and deep learning technology. As there are significant differences in the visual condition of every weather. It extracts the various information from the image and detects the weather condition based on the photometric property of the image such as color and texture of the image. There are generally two type images to deal with one which contain static weather conditions like sunny, hazy, fog and cloud and the other one which deals with the dynamic weather conditions like raining, snowing dust storm and hailstorm.



II. Related Work

In recent times, many contributions have been done in this field. Most of the related work has significance relevance to traffic surveillance and predicating the ideal condition to travel. Generally, all the methods focus on two aspects, weather condition and visual condition, but both aspects were considered different. But what we found out is that they are to be studied as co related fields. According to us visual conditions are significant changes to the appearance of the landscape cities and road. The images we are focused on and which we are considering will have different visual settings. For example, day and nighttime or dawn/dusk. We have observed the data from It's a well described data set which contains images from the different conditions. There are 10000 images which contain the related labels mapped to it. We have examined their images and work they did. We used this well designed and described data set form our image recognition project.

III. Methodology

As stated earlier, the aim was to detect weather conditions via images and classify into six categories: sunny, hazy, fog, cloud, raining, snowing, dust storm and hailstorm. We have

mostly focused on implementing CNN, Random forest and Decision tree algorithms. We then compared each model with the other model based on some metrics to find the best way to implement our model. We are taking input of the image and extracting six features namely mean and standard deviation for all the three channels. We will store the label as well, we divide the image set into two parts where 75% will be used to train the method and 25 % for testing the algorithm. We use a label encoder to encode the labels with numbers rather than text, for better accuracy. For displaying the results, we are using the classification_report for the sklearn.metrics module. The classification_report generates precision, recall and accuracy and computes a weighted accuracy for all the metrics and displays it.

We are using a dataset from the Visual Computing Research Center which consists of 10000 images with their labels for each weather type.[9]

A. KNN

In this algorithm we rely on the distance between the vectors. Kth nearest neighbor algorithm assumes that the vectors with the same properties exist near to each other. In the algorithm we are finding similar data points by calculating distance between them. The images are divided into data points and the same category image will be relative to closer datapoints. We have used K neighbors classifier from scikit- learn, passed the images with its labels. As compared to CNN it was low in accuracy but is relatively easy to implement, it has only two parameters which are required to implement KNN i.e. the value of K and the distance function. However, the accuracy of showing thunder images is particularly high but the precision for the other classification is less than 50%. The accuracy reduces as the size of the dataset becomes large.

```
[INFO] using 'knn' model
[INFO] evaluating...
              precision    recall  f1-score   support

      cloudy       0.33      0.34      0.34      2497
        haze       0.33      0.34      0.34      2454
       rainy       0.41      0.44      0.42      2509
        snow       0.42      0.44      0.43      2559
       sunny       0.43      0.39      0.41      2493
     thunder       0.76      0.74      0.75      2483

    accuracy                           0.45     14995
   macro avg       0.45      0.45      0.45     14995
weighted avg       0.45      0.45      0.45     14995
```

B. Decision Tree

It is a supervised machine learning technique, It is non parametric and is used for image classification. It is loaded with a series of tests that determine the label of the image. The tests are organized in the hierarchical structure, which is a decision tree, it basically works in the principle of divide and conquer and recursive partitioning. It divides the sets which are further divided into subsets, and then it checks for the homogenous nature of that set. Decision tree classifier module of scikit learn is used in the implementation of this algorithm. The result we got was not highly accurate as of CNN but each trial was faster than that of CNN. Max depth and min samples leaf were the parameters which highly affects our results. It should not be too high or low, and depending on the problem the field is set to give an efficient solution.

```
[INFO] using 'decision_tree' model
[INFO] evaluating...
              precision    recall  f1-score   support

      cloudy       0.30      0.28      0.29      2563
        haze       0.30      0.32      0.31      2474
       rainy       0.39      0.41      0.40      2457
        snow       0.41      0.40      0.41      2491
       sunny       0.38      0.37      0.38      2496
     thunder       0.70      0.69      0.69      2514

    accuracy                           0.41     14995
   macro avg       0.41      0.41      0.41     14995
weighted avg       0.41      0.41      0.41     14995
```

C. Random Forest

It is a supervised learning method, the basic building blocks of random forest are decision trees, i.e it is the collection of the decision trees. It builds multiple decision trees and merges them to get the more accurate and efficient solution. It follows a bagging method. While growing the trees it searches for the best feature among the random subset. This results in diversity which generally yields better results. In our project a random forest classifier of the scikit learn is used. While performing we have also used our previous experience of decision trees to make the right choices, as a result the accuracy of every category goes up on an average by 10-15%.

```
[INFO] using 'random_forest' model
[INFO] evaluating...
              precision    recall  f1-score   support

      cloudy       0.38      0.39      0.38      2530
        haze       0.42      0.35      0.38      2485
       rainy       0.45      0.53      0.49      2462
        snow       0.51      0.54      0.52      2509
       sunny       0.53      0.47      0.50      2495
     thunder       0.78      0.81      0.80      2514

    accuracy                           0.52     14995
   macro avg       0.51      0.52      0.51     14995
weighted avg       0.51      0.52      0.51     14995
```

D. Naive-Bayes

Naive Bayes is another supervised learning algorithm which looks at conditional probability distribution. This machine learning algorithm learns by looking at the training sets. The training set is nothing but the examples, which is already defined with the sets of the attribute. It is a probabilistic classifier It looks at the probability of assigning a class given a specific set of features.So for a given image we try to predict the weather given the features we extracted from the image, i.e the mean and standard deviation of the RGB channels. It will assume that each feature is independent of each other. We have used Gaussian NB from scikit learn to implement that in our project. It is the simplest algorithm to implement but also highly inefficient, it focuses on the target which will have a set of independent predictors, but in cases there are too many predictors and also it's very difficult to get a set of predictors which are completely independent. The precision of thunder is bit on the higher side as it has the entirely different predictors but the prediction of others classification are mostly around in the range of 25-35% which is quite inefficient

```
[INFO] using 'naive_bayes' model
[INFO] evaluating...
              precision    recall  f1-score   support

      cloudy       0.29      0.23      0.25      2462
        haze       0.38      0.09      0.15      2534
       rainy       0.29      0.37      0.33      2510
        snow       0.38      0.55      0.45      2441
       sunny       0.32      0.36      0.34      2513
     thunder       0.58      0.66      0.62      2535

    accuracy                           0.38     14995
   macro avg       0.37      0.38      0.36     14995
weighted avg       0.37      0.38      0.36     14995
```

E. Convolutional Neural Networks(CNN)

The task of image classification using CNN involves reading an input image, converting it into an array of pixels for the computer to understand and then passing that array through a series of convolutional layers. The output would be a list of probabilities of the image belonging to a specific class.Let us look into how each CNN layer works:First Layer:The first layer takes the input image and applies a filter on it. The filter convolves over the input image such that it multiplies the pixel values of the image with values in the filter to give us a single value which is stored in an activation map. Each value in the activation map represents an array of pixels in the input image. Understanding the filter from a higher-level perspective we can say that each filter acts as a feature identifier. For example, a filter can be using an edge detector or a curve line detector. Other layers: Between each convoluted layer there are many layers like ReLu activation layers through which the input image passes. These layers add nonlinearity to the input. Each layer adds more complexity to the input as it passes through them. For example, we can get a combination of features from previous layers to give us more complex shapes like square and circle. Fully Connected Layer: The fully connected layer finally takes the output from the previous layers as the input and produces an N dimensional vector which contains probabilities of each of the N classes. It correlates all the high-level features from the previous outputs and crates probabilities of each class.

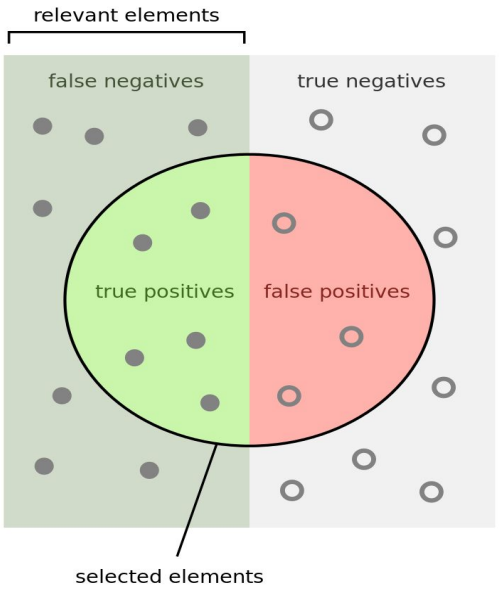
For our implementation we have resized the images to 32x32. This would give the 3072(32*32*3) inputs for the first layer. And we have used 2 hidden layers with ReLu. We can see that though it struggles a bitwith cloudy and haze weather, accuracy and precision layers are higher compared to other techniques across the board.

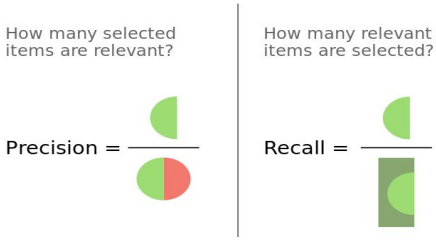
```
44984/44984 [==============================] - 93s 2ms/step
[INFO] evaluating network...
              precision    recall  f1-score   support

      cloudy       0.47      0.53      0.50      2510
        haze       0.68      0.63      0.65      2486
       rainy       0.62      0.57      0.59      2457
        snow       0.75      0.77      0.76      2501
       sunny       0.71      0.67      0.69      2520
     thunder       0.87      0.92      0.89      2521

    accuracy                           0.68     14995
   macro avg       0.68      0.68      0.68     14995
weighted avg       0.68      0.68      0.68     14995
```

IV. Metrics For Comparison

We are using Precision, Recall, F1 score and computing a weighted accuracy using all three. We are using the Weighted accuracy to compare the different machine learning techniques. A false positive is a result that indicates True, when it does not. A false negative is a result that indicates False, while in fact it does.

Precision is useful when we want low False Positives. Recall is useful when we want low False Negatives. F1 score combines both Precision and recall. It has a range of 0 to 1. F1 score is good when we want both False Positives and False Negatives to be low.





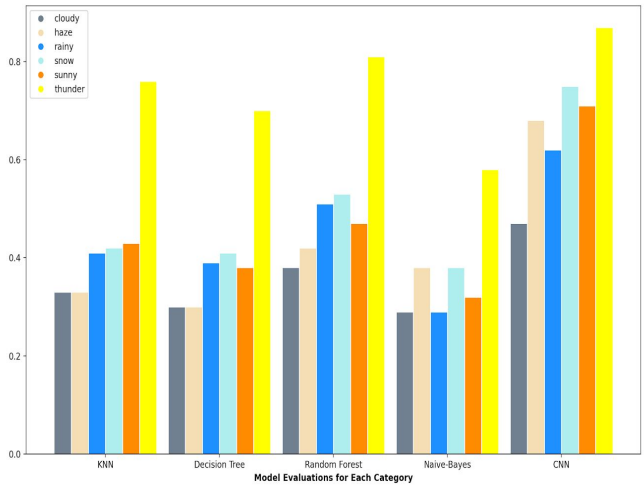
$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negetives}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

V. RESULTS

Model Evaluations for each weather condition :



As we have seen from the above results, the model built on CNN gave us the best results in terms of our metrics. The following are the images that our model built on CNN has predicted correctly :

But our model struggles to predict weather conditions such as cloudy, haze and snow. Below are a few images that our model has predicted wrongly. In general these images have a

lot of white pixels. As seen from the second image the model is not able to differentiate white clouds from snow.

Our model is also not able to predict correctly when the image represents multiple weather conditions such as the one below. The image was labelled cloudy in the dataset but our model thinks it is hazy.

References

[1] Q. Li, Y. Kong and S. Xia, "A method of weather recognition based on outdoor images," 2014 International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, 2014, pp. 510-516.

[2] Z. Zhu, L. Zhuo, P. Qu, K. Zhou and J. Zhang, "Extreme Weather Recognition Using Convolutional Neural Networks," 2016 IEEE International Symposium on Multimedia (ISM), San Jose, CA, 2016, pp. 621-625.

[3] M. Elhoseiny, S. Huang and A. Elgammal, "Weather classification with deep convolutional neural networks," 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, 2015, pp. 3349-3353.

[4] Z. Chen, F. Yang, A. Lindner, G. Barrenetxea and M. Vetterli, "How is the weather: Automatic inference from images," 2012 19th IEEE International Conference on Image Processing, Orlando, FL, 2012, pp. 1853-1856.

[5] Chu, W.-T., Zheng, X.-Y., Ding, D.-S., 2017. Camera as weather sensor: Estimating weather information from single images. J. Vis. Commun.

[6] Chu, W.-T., Zheng, X.-Y., Ding, D.-S., 2017. Camera as weather sensor: Estimating weather information from single images. J. Vis. Commun. Image Represent. 46, 233–249. https://doi.org/10.1016/j.jvcir.2017.04.002

[7] Hautiére, N., Tarel, J.-P., Lavenant, J., Aubert, D., 2006. Automatic fog detection and estimation of visibility distance through use of an onboard camera. Mach. Vis. Appl. 17, 8–20. https://doi.org/10.1007/s00138-005-0011-1

[8] https://www.pyimagesearch.com/2019/01/14/machine-learning-in-python/

[9] RSCM: Region Selection and Concurrency Model for Multi-class WeatherRecognition : http://vcc.szu.edu.cn/research/2017/RSCM.html