IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

USE OF DEEP LEARNING MODELS FOR WEATHER IMAGE PREDICTION

K. CHANDRASEKHAR [*1],
[*1]Assistant Professor, Computer Science & Engineering Department,
Anil Neerukonda Institute of Technology & Sciences,
Visakhapatnam, Andhra Pradesh, India.

ABSTRACT

Image data captured from outdoor visual devices are usually degraded by some external factors such as fog, rainfall, snow, smoke and haze. Hence the weather condition plays a prominent role in disturbing the public transportation systems as well as some advance assistance systems which try to capture the data from outdoor cameras and then function the system.Hence in order to avoid this disruption of images from outdoor cameras, one should able to identify the weather situation and what are the possible factors that may cause due to current weather images.In primitive days there is no such automatic method which can identify the weather images and process those images and find out the possible factors related to that weather image. Hence this motivated me to develop this proposed model, in which we try to collect some weather images and apply image processing techniques and CNN model to identify the weather image and its possible outcomes in order to take preventive steps from deweathering operations (e.g., removals of haze, rain, or snow) would be correctly triggered accordingly. Here we try to collect some weather images from jehanbhathena/weather-dataset and then apply VGG-19 as best CNN model to train the application and our experimental and simulation results clearly state that our proposed system achieved nearly 96 % of accuracy compared with other primitive models and hence this proposed VGG-19 is best and accurate in order to predict the weather possibility from weather images.

KEY WORDS:

CNN Model, VGG-19 Model, Weather Images, Image Processing, Public Transportation Systems, Assistance Systems.

1. INTRODUCTION

Tropical cyclones (TCs) are regarded as extreme weather events, along with gales, rainstorms, and storm surges, which can cause huge losses in coastal areas worldwide. In the past century, numerous meteorologists and warning centers devoted themselves to this study and made progress in observational technology, intensification physics; interactions of the atmospheric environment, the atmospheric boundary layer and air-sea interface, the ocean responses, and forecasting techniques [1]. However, many problems with predictive skills remain, particularly with the TC genesis, intensity, and risk forecasts. Generally, the most popular tropical cyclone dynamical forecast models have a relatively low accuracy, which is mainly due to the inaccurate vortex initialization of TCs, incomplete representation of complex physical processes, and coarse resolution [2,3]. There are studies that show that insufficient representations of the air–sea energy exchange under very high wind speed conditions would hinder simulating the intensity of TCs more effectively [4].

In addition, there is also a clear view that upper ocean feedback has important effects on TCs, but few operational numerical forecast models take it into consideration, which also reduces the performance of the models [4,5]. Additionally, other methods, such as statistical models, also are unable to deal with the complex and nonlinear relationship between TC-related predictors; thus, their forecast results need to be further improved [6–10]. In order to solve these problems of traditional methods, scientists began to consider using machine learning (ML) to explore satellite, radar, in-situ data, etc. to improve the forecast skills of TCs in recent years. Machine learning algorithms, as a means of artificial intelligence (AI), can be divided into three categories according to their applications: feature selection, clustering, and regression or classification [11].

Feature selection algorithms can eliminate irrelevant attributes through attribute selection to increase the effectiveness of the tasks, and then improve the accuracy of the models. For example, a typical Tucker decomposition method can solve the spatio-temporal problems that the traditional tensor decomposition algorithm cannot deal with [12]. A clustering algorithm is one of the earliest methods used in pattern recognition and data mining and can automatically divide a sample dataset into multiple categories. This has a wide range of applications in big data analysis.

Typical clustering algorithms include the finite-mixed model (FMM) [13], hierarchical clustering [14], and K-means algorithm [15]. As for classification or regression, one representative algorithm is support vector machine (SVM) for classification [16] and support vector regression (SVR) for regression [17], which can effectively deal with nonlinear problems by defining kernel functions. In addition, decision tree (DT) [18] is another typical algorithm that can mine and display the rules of classification, with high accuracy. A majority of works done with those mapping tasks are well performed with artificial neural networks (ANNs), which are considered as universal approximations for complex nonlinear mappings [12]. Since Hinton, a leading scholar of machine learning, put forward the deep neural network model in 2006, a new era of deep learning was opened.

2. LITERATURE SURVEY

In this section we will mainly discuss about the background work that is carried out in order to prove the performance of our proposed Method. Literature survey is the most important step in software development process. For any software or application development, this step plays a very crucial role by determining the several factors like time, money, effort, lines of code and company strength. Once all these several factors are satisfied, then we need to determine which operating system and language used for developing the application. Once the programmers start building the application, they will first observe what are the pre-defined inventions that are done on same concept and then they will try to design the task in some innovated manner.

MOTIVATION

SI NO.	TITLE OF THE PAPER	AUTHOR NAME AND YEAR OF PUBLICATION	TECHNIQUE USED	MERITS	DEMERITS
1	Image Recognition and Classification	B. Javidi,2012	Here we use ML algorithms for classification and recognition of several images.	Less computational complexity (execution time, memory) using ML algorithms	It is unable to classify for large datasets and also failed to classify for huge size images.
2.	Classification of Weather Phenomenon From Images by Using Deep Convolutional Neural Network	Haixia Xiao,Feng Zhang,2021	This paper is mainly used to identify the weather phenomenon based on images collected in various climatic conditions and we use CNN	The proposed RESNET model is used to classify the weather situations based on several images.	It is not giving more accuracy for large dataset.

			model.		
3.	Multi-Class Images for Weather Classification	Muhammad Hameed Siddiqi, 2021	Here we apply Multi class classifiers to classify the weather images.	Very accurate in comparison of several weather images and classify them easily. Easy to test on weather images to predict the future possibility	This Multi Class takes more time to process the weather images and hence time complexity should be reduced in future models.

3. EXISTING METHODOLOGY

In the existing system there was no proper method to classify the images and identify the name of that image category very efficiently and accurately, hence following are the main limitations in the existing system.

LIMITATION OF EXISTING SYSTEM

1. More Time Delay in finding the image and its category.
2. There is no technique which can classify all the images accurately.
3. There is no technique which can classify all the images as per dimensions.
4. All the existing methods try to classify images using ML algorithms

4. PROPOSED MODEL

Deep learning, in particular Convolutional Neural Network (CNN), is achieving significant results in image classification. In this paper a deep neural network model is been implemented which takes the images and it is then classified into the respective classes as per the dataset. Deep learning is one of the accurate and most reliable methods compared to the other methods. This paper presents a new technique to intensify the performance of image classification.

ADVANTAGES OF PROPOSED SYSTEM

1) By using CNN model it takes less time for the classification of weather images with more accuracy.

2) In this paper we survey different papers in which one or more algorithms of data mining used for the classification of weather images.

3) Result from using neural networks is nearly 100 % in this paper which can classify weather images accurately.

5. PROPOSED MACHINE LEARNING ALGORITHMS

The proposed system contains CNN Models and we try to compare two models in order to classify the weather images and then try to predict the future occurrence based on weather images. There are totally 4 modules present in this current application :

1) Gathering data,
2) Data Pre-Processing
3) Apply CNN Models
4) Data Interpretation

The whole approach is depicted by the following flowchart.

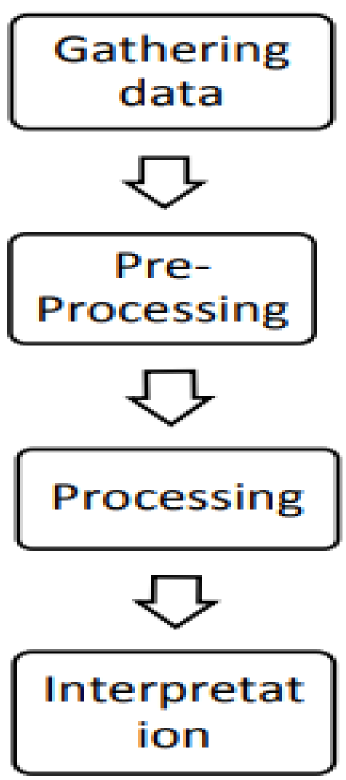


Figure. 1 Flowchart of the Technique

1) DATA GATHERING

Here we try to load the data set from kaggle and once dataset is downloaded we try to load the dataset to the system for performing the operations.

2) DATA PRE-PROCESSING

Data pre processing is a technique that is used to convert raw data into a clean dataset. The data is gathered from different sources is in raw format which is not feasible for the analysis. Pre-processing for this approach takes 4 simple yet effective steps.

Training and Test data: Splitting the Dataset into Training set and Test Set Now the next step is to split our dataset into two. Training set and a Test set. We will train our machine learning models on our training set, i.e our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to examine how accurately it will predict. A general rule of the thumb is to assign 80% of the dataset to training set and therefore the remaining 20% to test set.

3) APPLY CNN MODEL

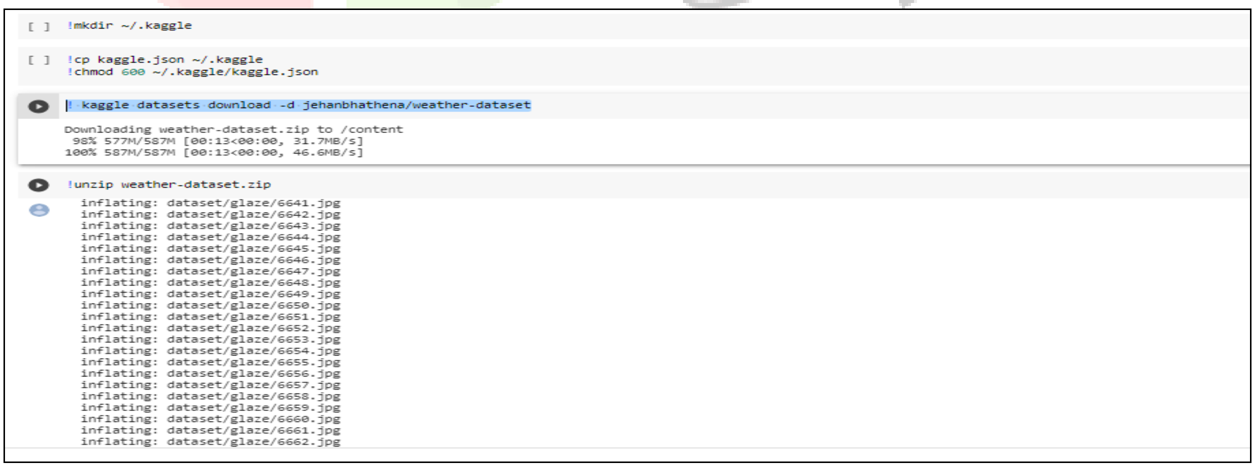
Once data is divided into test and train folders now we can apply well known CNN models and then check which model gives accurate and effective results on weather image classification dataset.

4) INTERPRETATION

The data set used for is further spitted into two sets consisting of two third as training set and one third as testing set. Here we apply VGG-19 and Inception V3 as CNN model to classify the images and display the name of that image based on its dimensions.

6. RESULT AND DISCUSSION

LOAD DATASET



UNZIP THE DATASET

```
!unzip weather-dataset.zip
  inflating: dataset/glaze/6603.jpg
  inflating: dataset/glaze/6604.jpg
  inflating: dataset/glaze/6605.jpg
  inflating: dataset/glaze/6606.jpg
  inflating: dataset/glaze/6607.jpg
  inflating: dataset/glaze/6608.jpg
  inflating: dataset/glaze/6609.jpg
  inflating: dataset/glaze/6610.jpg
  inflating: dataset/glaze/6611.jpg
  inflating: dataset/glaze/6612.jpg
  inflating: dataset/glaze/6613.jpg
  inflating: dataset/glaze/6614.jpg
  inflating: dataset/glaze/6615.jpg
  inflating: dataset/glaze/6616.jpg
  inflating: dataset/glaze/6617.jpg
  inflating: dataset/glaze/6618.jpg
  inflating: dataset/glaze/6619.jpg
  inflating: dataset/glaze/6620.jpg
  inflating: dataset/glaze/6621.jpg
  inflating: dataset/glaze/6622.jpg
  inflating: dataset/glaze/6623.jpg
  inflating: dataset/glaze/6624.jpg
```

DISPLAY CATEGORY OF NAMES

```
class_names=os.listdir(img_path)
class_names
```

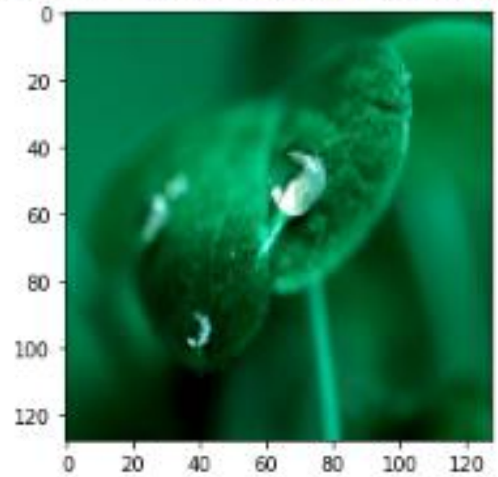
```
['rainbow',
 'snow',
 'hail',
 'fogsmog',
 'sandstorm',
 'lightning',
 'glaze',
 'rime',
 'rain',
 'frost',
 'dew']
```

```
[ ] def img_to_data(path):
      data=[]
      labels=[]

      for folder in sorted(class_names):
        full_path=os.path.join(path,folder)
        label=class_names.index(folder)
        for file_name in sorted(os.listdir(full_path)):
          try :
            file_path=os.path.join(full_path,file_name)
            img=cv2.imread(file_path)
            img=cv2.resize(img,(img_size,img_size),cv2.IMREAD_COLOR)
            data.append(img)
            labels.append(label)
          except Exception as e:
                print(e)
```

PLOT SAMPLE IMAGE

```
dew
<matplotlib.image.AxesImage at 0x7f2c5e95a290>
```



APPLY VGG-19 MODEL

```
from tensorflow.keras.applications.vgg19 import VGG19
from tensorflow.keras.layers import *
from tensorflow.keras.models import *

vgg=VGG19(weights='imagenet',include_top=False,input_shape=(img_size,img_size,3))
for layer in vgg.layers[:-1]:
  vgg.trainable=False

model=Sequential()
model.add(vgg)
model.add(Flatten())
model.add(Dense(len(class_names),activation='softmax'))

model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

model.summary()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels_notop.h5
80142336/80134624 [==============================] - 2s 0us/step
80150528/80134624 [==============================] - 2s 0us/step
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg19 (Functional)          (None, 4, 4, 512)         20024384

 flatten_1 (Flatten)         (None, 8192)              0

 dense_1 (Dense)             (None, 11)                90123
```

APPLY INCEPTION V3 MODEL

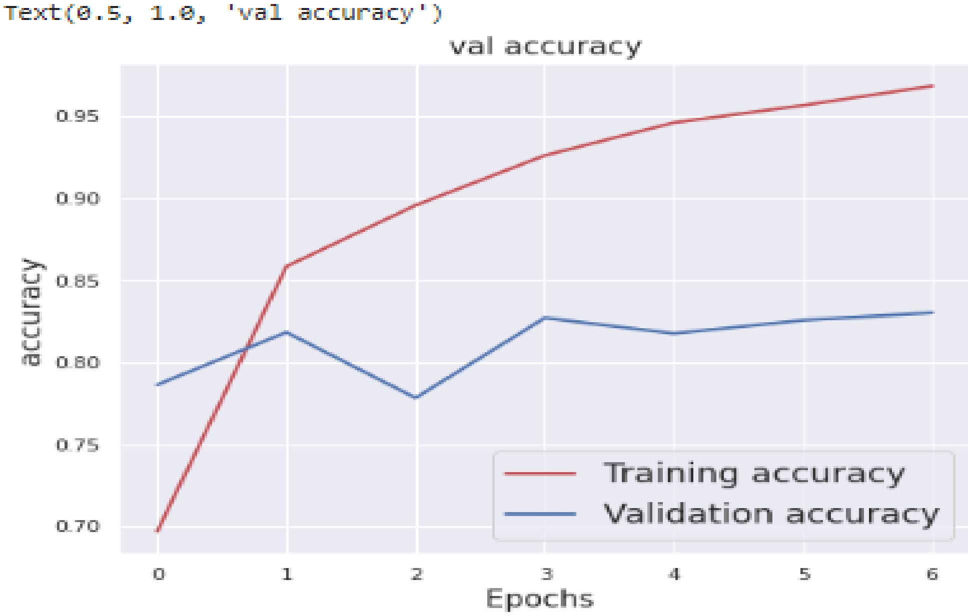
```
from tensorflow.keras.applications.inception_v3 import preprocess_input
X = preprocess_input(data)
print(X.shape)
X[0]
```

```
(6860, 128, 128, 3)
array([[[-0.99215686, -0.06666666, -0.40392154],
        [-1.        , -0.08235294, -0.41960782],
        [-1.        , -0.0745098 , -0.41176468],
        ...,
        [-0.9843137 , -0.17647058, -0.62352943],
        [-0.99215686, -0.18431371, -0.6313726 ],
        [-0.9843137 , -0.17647058, -0.62352943]],

       [[-0.99215686, -0.06666666, -0.38823527],
        [-1.        , -0.0745098 , -0.41176468],
        [-0.9843137 , -0.08235294, -0.3960784 ],
        ...,
        [-0.99215686, -0.19215685, -0.62352943],
```

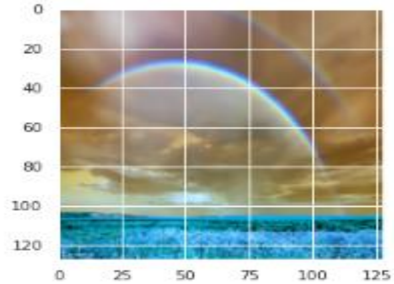
TRAIN AND TEST VALIDATION

```
Text(0.5, 1.0, 'val accuracy')
```



PREDICT CLASS NAME FROM SAMPLE INPUT IMAGE

```
(1, 128, 128, 3)
```




```
[ ] pred=model.predict(x1) # for predicting class
    labels_pred=np.argmax(pred,axis=1)
    labels2=class_names[labels_pred[0]]
    pred_results=pd.DataFrame(data=pred,columns=class_names)
    fig=plt.figure(figsize=(10,8))
    sns.set_theme(style="darkgrid")
    ax=sns.barplot(data=pred_results)
    ax.set_xticklabels(class_names,rotation=90)
    ax.set_xlabel('Class')
    ax.set_ylabel('Accuracy')
    ax.set_title('Predicting class name ')
    plt.show()
```

7. CONCLUSION

Tropical cyclones have been a concern of meteorologists for more than 100 years. Numerous scholars have conducted in-depth studies on key issues, such as the structure, dynamics, and forecasting techniques. Machine learning is derived from statistical methods that can automatically discover relevant rules from large-scale data for detection, analysis, prediction, etc. The application of machine learning for the key problems of TCs provides a new way of thinking to address many bottlenecks in this field. Techniques based on a pure data-driven approach and using machine learning to improve numerical models have both been shown by a large number of studies to provide huge contributions to improving TC predictions. Although existing research has made some progress in genesis forecasts, path prediction, intensity prediction, TC weather prediction, and improving numerical forecast models by integrating machine learning, there are still many aspects that remain to be studied, which we regard as both an opportunity and a challenge.

.

8. REFERENCES

1. Emanuel, K. 100 Years of Progress in Tropical Cyclone Research. Meteorol. Monogr. 2018, 59, 15.1–15.68.

2. Ma, L.-M. Research Progress on China typhoon numerical prediction models and associated major techniques. Prog. Geophys. 2014, 29, 1013–1022.

3. Tropical Cyclone Forecast Model. Available online: https://en.wikipedia.org/wiki/Tropical_cyclone_forecast_model (accessed on 1 May 2020).

4. Moon, I.J.; Ginis, I.; Hara, T.; Thomas, B. A Physics-Based Parameterization of Air-Sea Momentum Flux at High Wind Speeds and Its Impact on Hurricane Intensity Predictions. Mon. Weather Rev. 2007, 135, 2869–2878.

5. Bender, M.A.; Ginis, I.; Kurihara, Y.J. Numerical simulations of tropical cyclone-ocean interaction with a high-resolution coupled model. J. Geophys. Res. Atmos. 1993, 98, 23245–23263.

6. Lee, C.Y.; Tippett, M.K.; Camargo, S.J.; Sobel, A.H. Probabilistic Multiple Linear Regression Modeling for Tropical Cyclone Intensity. Mon. Wea. Rev. 2015, 143, 933–954.

7. Wang, Y.; Rao, Y.; Tan, Z.M.; Schönemann, D. A Statistical Analysis of the Effects of Vertical Wind Shear on Tropical Cyclone Intensity Change over the Western North Pacific. Mon. Weather Rev. 2015, 143, 3434–3453.

8. DeMaria, M.; Kaplan, J. An updated statistical hurricane intensity prediction scheme (SHIPS) for the Atlantic and eastern North Pacific basins. Weather Forecast. 1999, 14, 326–337.

9. DeMaria, M.; Kaplan, J.J.W. A statistical hurricane intensity prediction scheme (SHIPS) for the Atlantic basin. Weather Forecast. 1994, 9, 209–220.

10. Demaria, M.; Mainelli, M.; Shay, L.K.; Knaff, J.A.; Kaplan, J. Further Improvements to the Statistical Hurricane Intensity Prediction Scheme (SHIPS). Weather Forecast. 2005, 20, 531–543.

11. Machine Learning. Available online: https://en.wikipedia.org/wiki/Machine_learning (accessed on 9 April 2020).

12. Kim, Y.D.; Choi, S. Nonnegative tucker decomposition. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1–8.

13. Melnykov, V.; Maitra, R. Finite mixture models and model-based clustering. Stat. Surv. 2010, 4, 80–116.

14. Johnson, S.C. Hierarchical clustering schemes. Psychometrika 1967, 32, 241–254. [CrossRef]

15. Krishna, K.; Murty, M.N. Genetic K-means algorithm. IEEE Trans. Syst. Man Cybern. Part B (Cybernetics) 1999, 29, 433–439.

16. Suykens, J.A.; Vandewalle, J. Least squares support vector machine classifiers. Neural Process. Lett. 1999, 9, 293–300.

17. Drucker, H.; Burges, C.J.; Kaufman, L.; Smola, A.J.; Vapnik, V. Support vector regression machines. In Proceedings of the 9th International Conference on Neural Information Processing Systems; MIT Press: Denver, CO, USA, 1997; pp. 155–161.

18. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. IEEE Trans. Syst. 1991, 21, 660–674.