Deep Learning – Case Study

Title: Weather Analysis using CNN

Name: Kantariya Manthan Jayendra

Enrollment Number: 18012011032

Batch: DL1

Guided by: Prof Ketan J. Sarvakar

1. Abstract

Weather forecasting has gained attention many researchers from various research communities due to its effect to the global human life. The emerging deep learning techniques in the last decade coupled with the wide availability of massive weather observation data and the advent of information and computer technology have motivated many researches to explore hidden hierarchical pattern in the large volume of weather dataset for weather forecasting. This study investigates deep learning techniques for weather forecasting. In particular, this study will compare prediction performance of Recurrence Neural Network (RNN), Conditional Restricted Boltzmann Machine (CRBM), and Convolutional Network (CN) models. Forecasting accuracy of each model is evaluated using Frobenius norm. The result of this study expected to contribute to weather forecasting for wide application domains including flight navigation to agriculture and tourism.

2. Problem Description

Most of the country is analyzing the weather to be ready for the coming time weather. So by analyzing the previous weather data, they can be ready if there is bad weather in future that can cause harm like cyclone, flood etc. In this project, I will train a Deep Learning model for weather analysis using Python. I will use the Tensor flow and Keras library in Python for weather analysis model. For making a weather analysis model, I will train a Convulation Neural Network (CNN).

3. Notebook Used

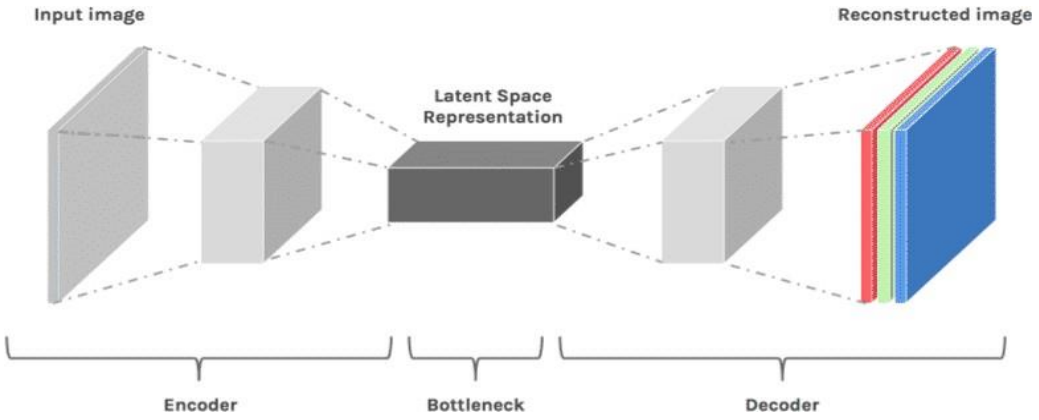
Google Colab - Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

4. Tools and Libraries

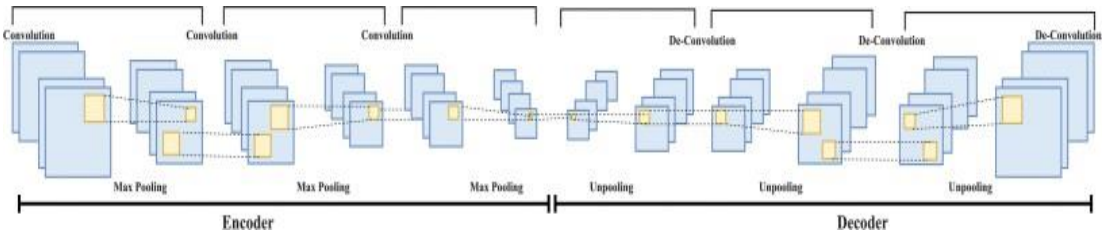
Tools and Libraries	Usage
Keras	This library is used for building the network architecture. It allows us to use several layers, callbacks, and InceptionResNetV2 model.
SkImage	Library provides various image transformation, colors, cropping related functions.
Tqdm	Provides a good progress bar that can be combined with a loop to get visualization of the current progress.
PIL	For loading the viewing the images.
Numpy	We are using it for the Image matrix handling.

5. Model Explanation and Architecture

The model architecture is a combination of both AutoEncoders and ResNet classifier. The Encoder at its best is just shades everything in a brownish tone. Hence, to let the network an "idea" about what things to color, we add ResNet.



The InceptionResNetV2 model is loaded with pretrained weights available. These weights are taken from the model which was trained on nearly 1.2 million images. AutoEncoder contains three main parts: 1. Encoder, Fusion, Decoder parts.



Encoder holds the Conv2D and MaxPooling2D Layers. Fusion holds the RepeatVectorand concatenation with the previous output layers. Decoder holds the Conv2D and UpSampling2D layers.

6. WorkFlow

Step 1: Import Requires Libraries

Step 2: Visualize the Data

Step 3: Preprocess data function

Step 4: Train data

Step 5: Load the dataset in x and y format

Step 6: Split data into train and test split

Step 7: Design CNN model

Step 8: Visualize CNN model

Step 9: Transfer Learning

Step 10: Load transfer learning model

Step 11: Transfer learning model visualization

Step 12: Result Prediction

7. Dataset Name & Link

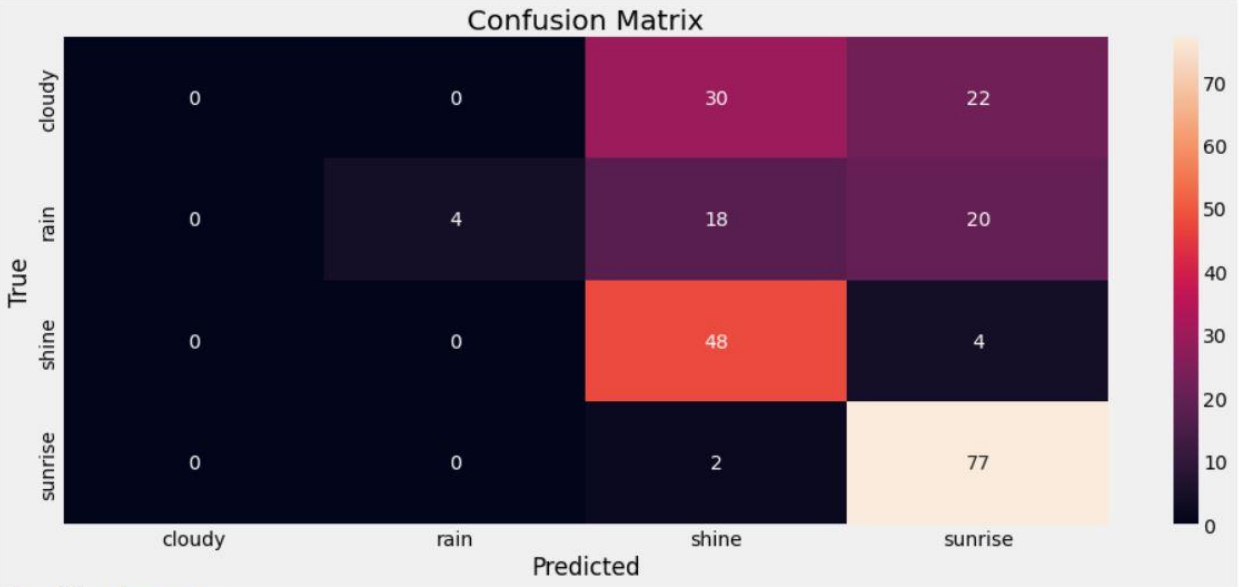
Weather_data.zip

https://drive.google.com/u/3/uc?id=1VmG3JJxqeipRhbIt-O0E92r1nDuj7P9z

8. Github Link

https://github.com/manthan181100/DL_Project

9. Output



Classification Report

	precision	recall	f1-score	support
cloudy	0.00	0.00	0.00	52
rain	1.00	0.10	0.17	42
shine	0.49	0.92	0.64	52
sunrise	0.63	0.97	0.76	79
accuracy			0.57	225
macro avg	0.53	0.50	0.39	225
weighted avg	0.52	0.57	0.45	225