

EXPLORING THE INTERSECTION OF MACHINE LEARNING AND IMAGE PROCESSING

Abstract

In this project, we explore how machine learning techniques can be applied to predicting season/weather from an image. The scope of this project is restricted to the image given as input. In this study, we propose and evaluate a deep neural-network based strategy. We have chosen CNN to be the most apt algorithm for our project's practical use case. By leveraging convolutional neural networks (CNNs), we take advantage of their ability to automatically learn meaningful representations directly from the input images. Through a supervised learning approach, we collect a substantial amount of labeled data, enabling us to train and fine-tune our model efficiently. In comparison to standard practices our results show remarkable improvement and the approach is still lightweight enough to run on modest computer systems. Furthermore, a minute extension has been added to the project and the user can obtain weather details of any city that is extracted through a weather Api.

Keywords: CNN, season prediction, deep neural network, image processing.

Authors

Paduru Ananya

Student
II Year CSE
Keshav Memorial Institute of Technology
Hyderabad, Telangana
India.

Isha Jain

Student
II Year CSE (AI & ML)
Keshav Memorial Institute of Technology
Hyderabad, Telangana
India.

Manvitha Chikoti

Student
II Year CSE (AI & ML)
Keshav Memorial Institute of Technology
Hyderabad, Telangana
India.

Chitimella Praneeth

Student
II Year CSE (AI & ML)
Keshav Memorial Institute of Technology
Hyderabad, Telangana
India.

Sushmitha Chigullapally

Student
II Year CSE (AI & ML)
Keshav Memorial Institute of Technology
Hyderabad, Telangana
India.

I. INTRODUCTION

Deep neural networks (DNNs) are composed of a multilayer architecture, enabling them to transform raw datasets from their original feature space into a learned feature space. This means that instead of manually selecting features, neural networks can "learn" these features, resulting in higher accuracy and improved generalization. Deep learning has yielded promising results in various domains, including computer vision (CV), speech recognition, natural language processing (NLP), as well as scientific disciplines such as Physics, Chemistry, and Bioinformatics. Notably, DL-based weather prediction has garnered significant attention from prestigious entities such as the European Centre for Medium-Range Weather Forecasts, the academic journal Nature, and various enterprises. This project is a multi-page website that focuses primarily on Image Processing (i.e.) analyzing the image given as input and displaying the relevant weather condition as output. Furthermore, the weather conditions of all major cities in the world can be extracted and displayed as well. [4,5,6]

II. METHODOLOGY

Convolutional Neural Networks are different from other neural networks (Neural networks is an AI method that thinks similar to that of a human brain) because of their ability to deal with images and audio signals. Thus, we have chosen to use this in our project. The three layers involved in CNN are:

- 1. Convolutional Layer:** The convolutional layer serves as the fundamental building block within a Convolutional Neural Network (CNN), playing a central role in most of the computation. It involves several essential components like: input data, a filter (also called a feature detector or kernel), and a feature map. In this context, let's consider the input as a color image, represented by a 3D matrix of pixels, where the dimensions correspond to height, width, and depth (representing the RGB channels). During the process of convolution, the filter moves across the receptive fields of the image, examining the presence of specific features within these regions. This way, the convolutional layer analyzes the input image to create meaningful feature maps, which are crucial for the subsequent stages of the CNN's operation.
- 2. Pooling Layer:** Pooling refers to a down sampling operation that plays a vital role in reducing the dimensionality of the input data, effectively decreasing the number of parameters. Similar to the convolutional layer, pooling involves applying a filter across the entire input. However, unlike the convolutional layer, the pooling filter does not have any learnable weights associated with it. Instead, the filter utilizes an aggregation function to process the values within its receptive field, generating the output array. The two primary types of pooling are: max pooling and average pooling.
- 3. Fully-Connected Layer:** The fully connected layer in CNN acts as a bridge between the spatial feature extraction layers and the output layer. In the previous layers (partially connected layers), the pixel values of the input image are not directly connected to the output layer. However, in the fully connected layer, each neuron receives input from all neurons in the previous layer, and a weighted sum of these inputs is computed, followed by the application of an activation function. The output of this layer is fed into a final layer, such as a SoftMax layer for classification tasks, where the probabilities for different

classes are computed. This allows CNN to make predictions based on the learned features. [7,8,10,11]

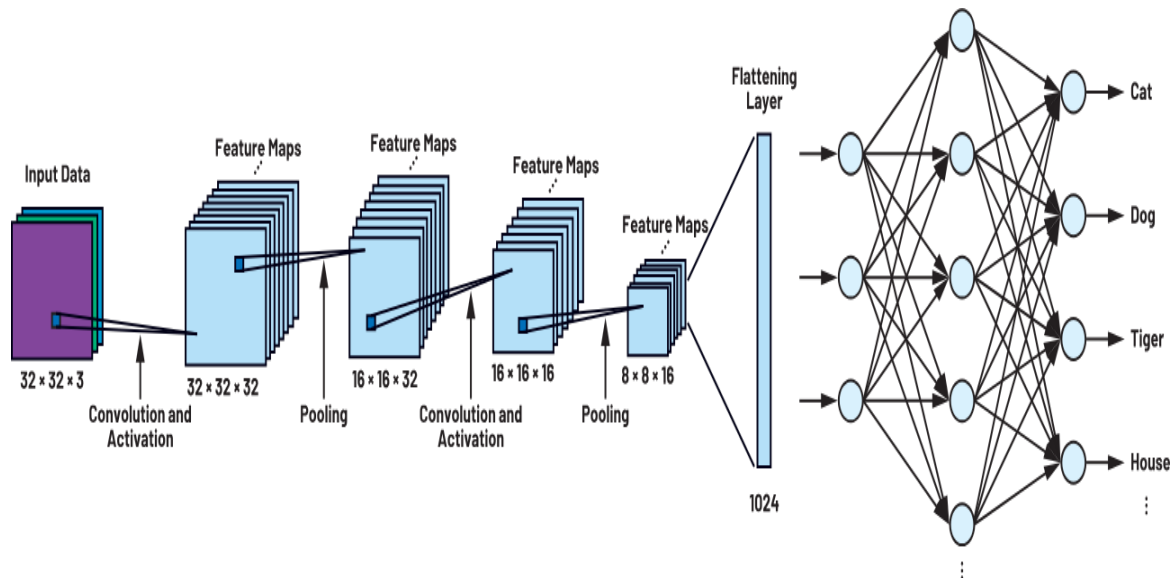


Figure 1: A CNN Model

III. DATASET AND FEATURES

The dataset has 1125 images related to weather that can be broadly classified into Cloudy, Shine, Rainy, Sunrise. After pre-processing of the dataset, Shine was labeled to Sunny. We also added Snowy (Winter) images to the dataset. Pre-Processing was performed and relevant pictures were chosen for training. [1,3]

IV. EXPERIMENTAL

Neural networks are preferred choice for a weather forecasting project as they handle nonlinear dependencies of past weather data and future weather conditions which is not the case with regression models. Deep learning algorithms like YOLO (You Only Look Once), SSD (Single Shot Multibox Detection), and deep reinforcement learning can also be implemented for image processing. Deep Reinforcement is an approach that combines deep learning with reinforcement learning to enable agents to learn decision-making policies. Algorithms like YOLO, SSD are preferred for real time object detection like video or live stream. However, CNN can do image classification, object detection and segmentation together in one model and has the upper edge. [7,8,10,11]

V. ALGORITHMS

Prior to choosing CNN, various algorithms like KNN, SVM, XGBoost were implemented using a csv dataset.

- 1. KNN:** It relies on the notion of proximity or distance between data points, but in high-dimensional spaces, the concept of distance becomes less meaningful. KNN, being a non-parametric algorithm, does not learn these spatial relationships and treats each pixel or feature independently, resulting in limited performance for image classification tasks.

- SVMs:** It can face challenges in handling high-dimensional data efficiently. Image classification often requires capturing patterns and structures that are invariant to translations, rotations, and scale changes. SVMs, without incorporating specific mechanisms to handle these variations, may struggle to capture such invariant features effectively. SVMs typically require solving a quadratic optimization problem involving the entire training set, which can become computationally expensive as the number of training samples increases. In contrast, CNNs can be trained more efficiently on large-scale image datasets using techniques like mini-batch gradient descent.
- XGBoost:** It treats each feature independently and does not inherently possess the ability to learn and leverage the spatial relationships between pixels of image patches.
- ANN:** ANNs do not allow parameter sharing across different spatial locations in an image. This reduces the number of learnable parameters compared to CNN. The accuracy for these models was less than 80% and other pre-trained models, in particular DenseNet and ResNet showed over fitting issue (high accuracy in training data but lower in test), whereas CNN's accuracy was the most idealistic.[2]

VI. IMPLEMENTATION

- Creating a dataset of various images of different seasons (Sunny, Rainy, Cloudy, Winter, Sunrise).**

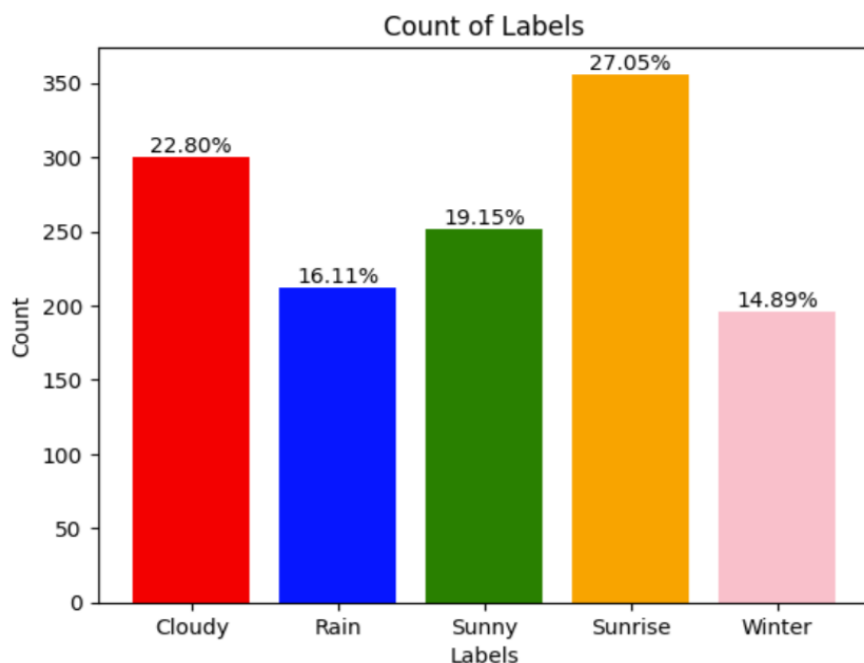


Figure 2: Bar Graph representing count of labels in each category.

2. Data Preprocessing and Performing Data Augmentation.



Figure 3: Various Augmented Images

- 3. Training Model:** Initially, the layers and architecture of the neural network are defined. In Convolutional Neural Network (CNN), the three layers used are convolutional layers, pooling layers, and a fully connected layer. Once the model is defined it is tested on the dataset for training and evaluation.

```

Model: "model"
-----
Layer (type)                Output Shape                Param #
-----
input_1 (InputLayer)        [(None, 224, 224, 3)]      0
conv2d (Conv2D)              (None, 224, 224, 32)       896
conv2d_1 (Conv2D)           (None, 224, 224, 32)       9248
max_pooling2d (MaxPooling2D) (None, 112, 112, 32)       0
conv2d_2 (Conv2D)           (None, 112, 112, 64)       18496
conv2d_3 (Conv2D)           (None, 112, 112, 64)       36928
max_pooling2d_1 (MaxPooling2D) (None, 56, 56, 64)       0
flatten (Flatten)           (None, 200704)              0
dense (Dense)                (None, 200)                 40141000
dropout (Dropout)           (None, 200)                 0
dense_1 (Dense)              (None, 100)                 20100
dense_2 (Dense)              (None, 5)                   505
-----
Total params: 40,227,173
Trainable params: 40,227,173
Non-trainable params: 0
-----

```



Figure 4: Model

4. Accuracy and Sample Prediction. [12,13,14]

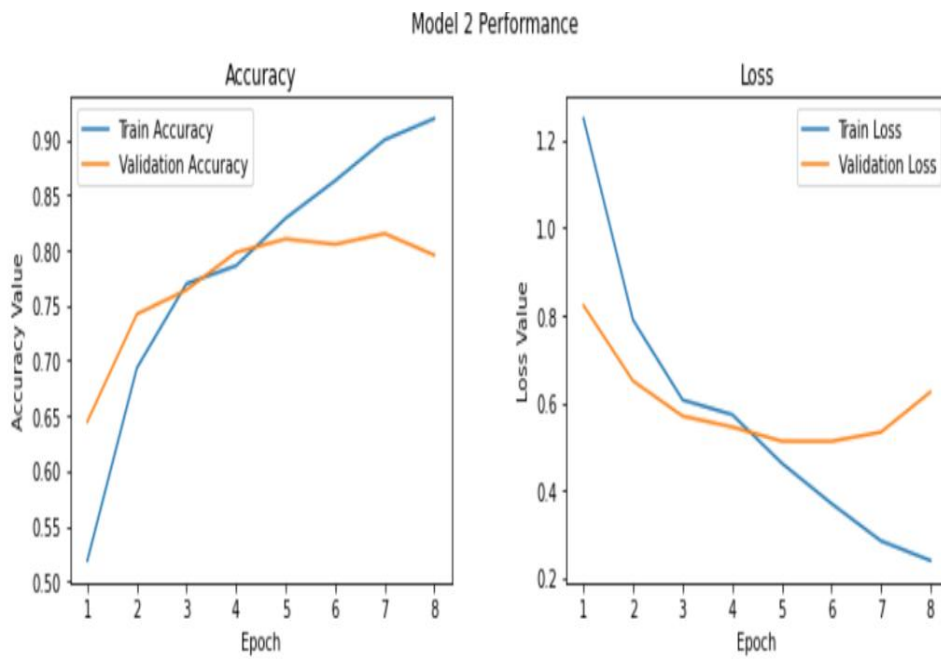


Figure 5: Graphs displaying performance of model

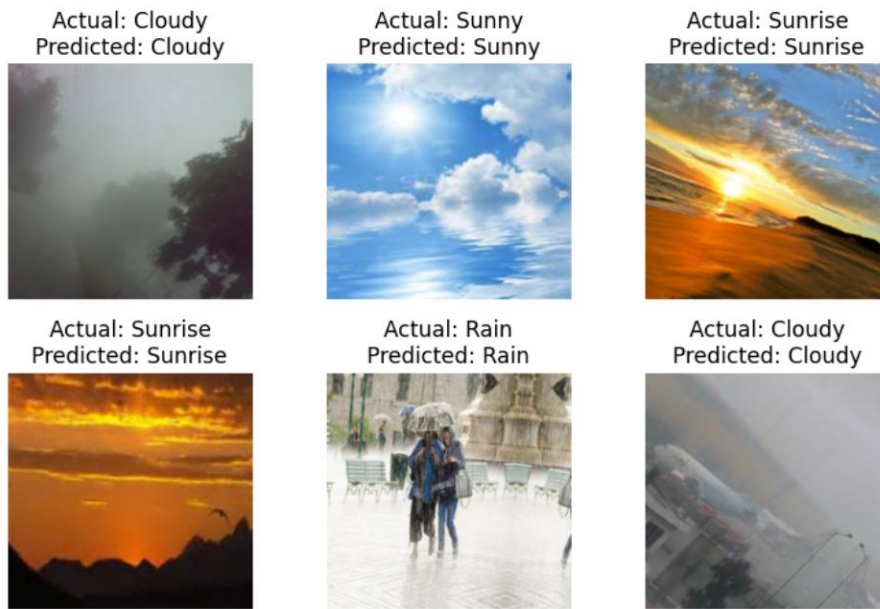


Figure 6: Images showcasing prediction of model

5. **Extracting weather details from an Api (openweather or rapidapi client) using Axios or fetch requests (used to make HTTP requests in JavaScript applications).[9,14,15,16]**



Figure 7: Sample image of weather details displayed for a city

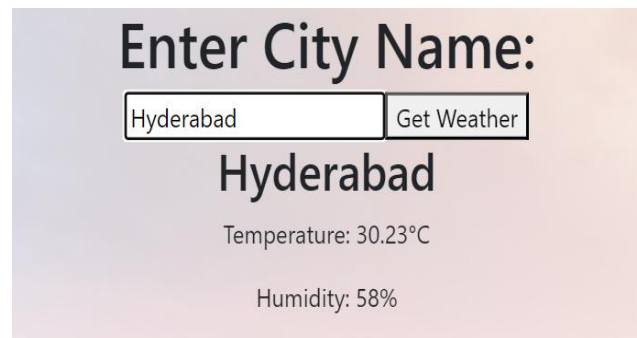


Figure 8: Sample image of weather details displayed for a city

VII. CONCLUSION

Algorithms like KNN, SVM, ANN, Decision Trees, XGB were outperformed by the CNN model. CNN uses a super-resolution procedure (i.e.) it learns from data in a supervised learning method, where labeled data is used. We conclude from our observations that there is a significant increase in performance when our proposed strategy is used as compared to other procedures. In addition to this, training the model can be done on the system and our model runs on a single GPU making it more economical.

VIII. ACKNOWLEDGMENTS

We express our sincere gratitude to the management of KMIT for their unwavering support throughout the process of writing this paper. We extend our heartfelt thanks to Prof. Neil Gogte sir and Mr. Venkat Rao sir for their invaluable inputs, guidance, and expertise. We are especially grateful to our guide and mentor, Mr. R.V. Gandhi, for his dedicated assistance and continuous support throughout this endeavor.

REFERENCES

- [1] <https://www.kaggle.com/datasets/pratik2901/multiclass-weather-dataset?select=Multi-class+Weather+Dataset>
- [2] <https://www.kaggle.com/code/tolulope123/weather-forecasting-project/input>
- [3] <https://www.kaggle.com/datasets/balraj98/summer2winter-yosemite> (Winter images)
- [4] <http://cs229.stanford.edu/proj2016/report/HolmstromLiuVo-MachineLearningAppliedToWeatherForecasting-report.pdf>
- [5] <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8588749> (2018 IEEE 14th International Conference on e-Science)
- [6] Xinyu Lei; Hongguang Pan; Xiangdong Huang IEEE Access >Volume: 7 : A Dilated CNN Model for Image Classification
- [7] https://www.researchgate.net/publication/252655682_Objective_definition_of_seasons_to_detect_their_changes_based_on_climate_data
- [8] <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.025643>
- [9] [https://api.openweathermap.org/data/2.5/weather?q=\\${city}&appid=\\${API_KEY}&units=metric](https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${API_KEY}&units=metric)
- [10] <https://cse.anits.edu.in/projects/projects1920B4.pdf>
- [11] Ivan Culjak; David Abram; Tomislav Pribanic; Hrvoje Dzapo; Mario Cifrek, A brief introduction to OpenCV, >2012
- [12] Proceedings of the 35th
- [13] <https://ieeexplore.ieee.org/document/6240859>
- [14] <https://keras.io/api/>
- [15] <https://flask.palletsprojects.com/en/2.3.x/>
- [16] <https://wsgi.readthedocs.io/en/latest/>
- [17] <https://rapidapi.com/>