Exploring the Intersection of Machine Learning and Image Processing

[1]Paduru Ananya, [2] Isha Jain, [3] Manvitha Chikoti, [4] Chitimella Praneeth, [5]Sushmitha Chigullapally
[1,2,3,4,5]II Year CSE Students, Keshav Memorial Institute of Technology

Abstract: Machine learning techniques are more robust to perturbations, in this project we explore their application to *predicting the season/weather* based on the image given as input. The scope of this project is restricted to the image given as input. In this project, we propose and evaluate a strategy based on a *deep neural network*. 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. Our results show significant improvement when compared with standard practices and the strategy is still lightweight enough to run on modest computer systems.

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

1. Introduction: The deep neural network (DNN) is composed of multilayer architecture, which can reconstruct the raw data sets from the original feature space into a learned feature space. In other words, they can "learn" features by neural networks instead of selecting features manually, and achieve higher accuracy and better generalization with the learned features. Deep learning has achieved encouraging results in many areas, such as computer vision , speech recognition, natural linguistic programming, as well as in scientific fields in physics, chemistry and bioinformation. DL-based weather prediction has attracted attention in many fields, such as the authoritative meteorological research institution European Centre for Medium-Range Weather Forecasts, academic journal Nature and enterprises.

Our 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]

2. Methodology: Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs and hence we chose to use this in our project. They have three main types of layers, which are:

2.1 Convolutional Layer

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let's assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions - a height, width, and depth - which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.

2.2 Pooling Layer

Pooling layers, also known as downsampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array. There are two main types of pooling: max pooling and average pooling.

2.3 Fully-Connected Layer

The name of the fully-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully connected layer, each node in the output layer connects directly to a node in the previous layer. [7,8,10,11]

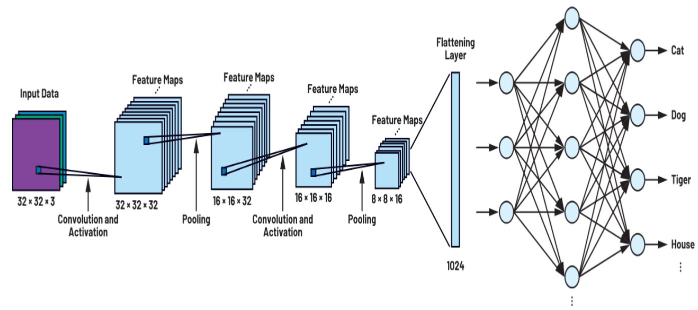

Figure 1: A CNN Model

3. 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]

4. Experimental: Neural networks seem to be the popular machine learning model choice for weather forecasting because of the ability to capture the non-linear dependencies of past weather trends and future weather conditions, unlike functional regression models used. Deep learning algorithms like YOLO (You Only LLook Once), SSD (Single Shot Multibox Detection), and deep reinforcement learning can also be implemented for image processing. However, CNN can do image classification, object detection, segmentation etc together in one model. While algorithms like YOLO, SSD are preferred for object detection mainly used for real time objects like video or live stream. Deep Reinforcement is an approach that combines deep learning with reinforcement learning to enable agents to learn decision-making policies. [7,8,10,11]

5. Algorithms:

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

5.1 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.

5.2 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.

5.3 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.

5.4 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%, whereas CNN's accuracy was greater.[2]

6. Implementation:

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

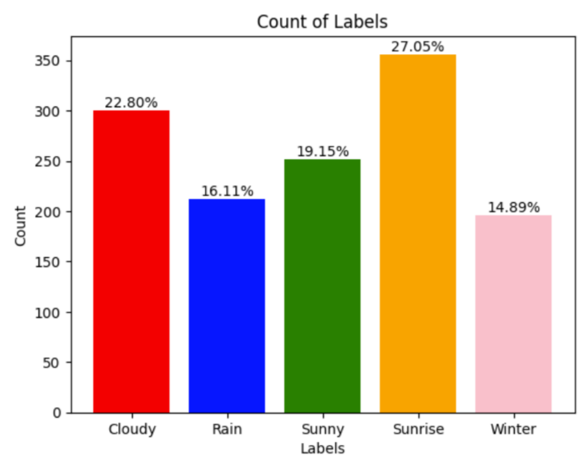

Figure 2: Bar Graph representing count of labels in each category.

6.1(b): Data preprocessing and performing Data Augmentation.

Figure 3: Various Augmented Images

6.1(c): Training Model: Initially, the layers and architecture of the neural network are defined. In the case of a Convolutional Neural Network (CNN), convolutional layers, pooling layers, and a fully connected layer is used. 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 (MaxPooling  (None, 56, 56, 64)       0
 2D)

 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

6.1(d): Accuracy and Sample Prediction.[12,13,14]

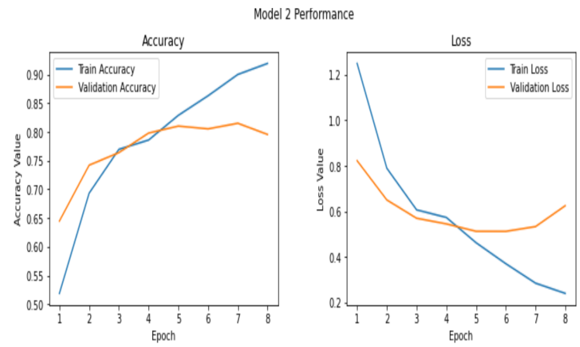

Figure 5: Graphs displaying performance of model

Figure 6: Images showcasing prediction of model

6.2: Extracting weather details from an api using axios request. [9,14]

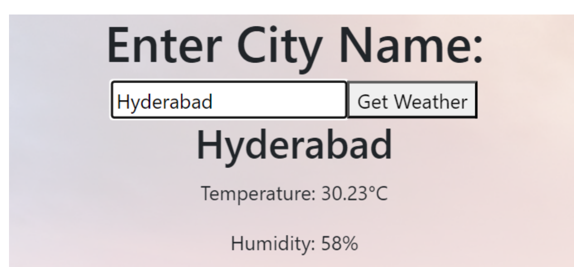

Figure 7: Sample image of weather details displayed for a city

7. Conclusion: Algorithms like KNN, SVM, ANN, Decision Trees, XGB were outperformed by the CNN model. This approach has a major advantage that the super-resolution procedure is learned from data in a supervised learning fashion. Moreover, there is no need to manually label the data. From our experiments, we observed a significant improvement of the proposed strategy compared with the standard downscale procedures. Moreover, the strategy is economical and can run in a single GPU system, and even training can be run on that system.

Acknowledgments: *We would like to 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
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 Proceedings of the 35th ...
12. https://ieeexplore.ieee.org/document/6240859
13. https://keras.io/api/
14. https://flask.palletsprojects.com/en/2.3.x/
15. https://wsgi.readthedocs.io/en/latest/
16. https://rapidapi.com/