

Deep Learning

Weather Image Recognition

Prepared by:

- Amgad Shalaby
- Ashraf Mahmoud
- Hamdy Hamada zaied
- Nada Ibrahim

Under the Supervision of:

- Eng. Abdelrahman A. Eid

Table of contents

01

Introduction

02

Approach

03

Results

04

Deployment

01

Introduction

Importance of Knowing Weather Conditions

- In various industries, the ability to accurately assess current and historical weather conditions is crucial for effective decision-making and operational optimization.
- For instance, farmers might look at the current weather to decide if the sprinklers should be turned on or off. Ski resort operators might choose to enable snowmaking machines based on varying weather conditions across the mountain. Construction workers might plan out the supplies and rain gear they'll need for a remote job site. Racing games like Formula One (F1) might deiced to change the tires of the car from the ordinary one to the full wet tires which used in heavy raining.



Traditional Way

Currently, making such decisions can require manually looking at video feeds from remote cameras, relying on weather forecasts, or simply looking out the window.



AI Way

- Using **deep learning (DL)** offers the potential to automate this by providing a **digital eye**.
- More specifically, if an image recognition DL model could be built to identify conditions by simply looking at images of the weather, it could be deployed in scenarios like those described above.
- For example, a camera feed on a farm could be processed by an DL model deployed on an IoT device at the edge (e.g., on a smart camera). That model can then be used to automatically determine the current weather conditions and enable or disable sprinkler valves accordingly.



Taking Action!!

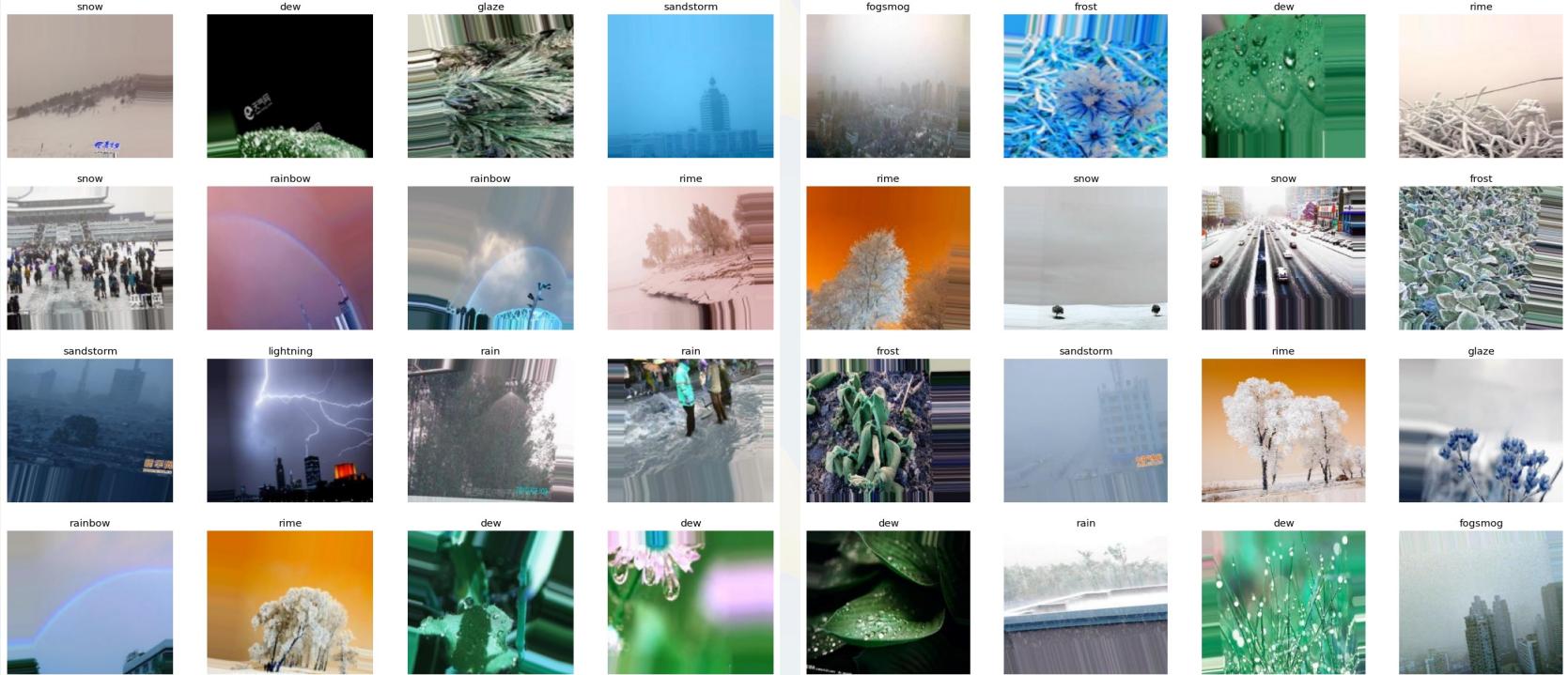
To demonstrate this use case, we built a model trained to classify 11 different types of weather. We used **6862** images from the [Multi-class Weather Dataset for Image Classification](#): dew, fog/smog, frost, glaze, hail, lightning , rain, rainbow, rime, sandstorm and snow.



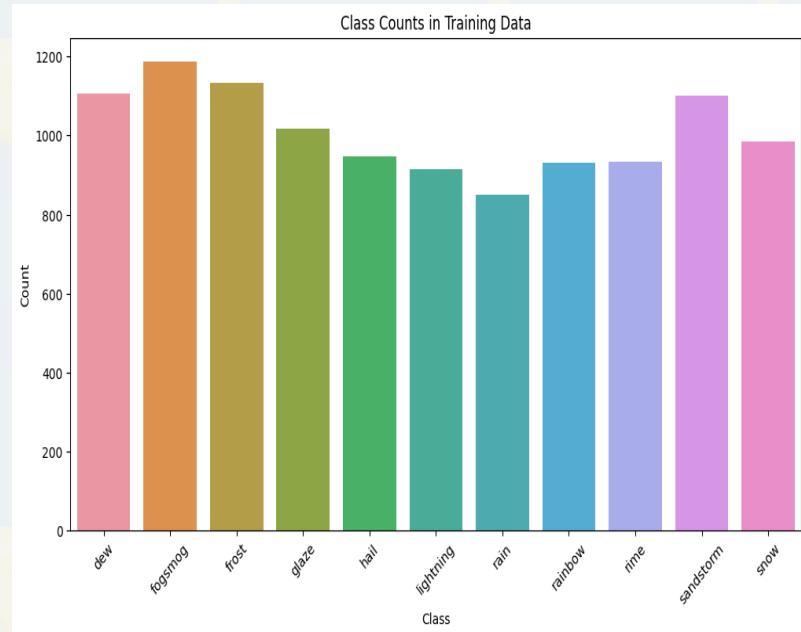
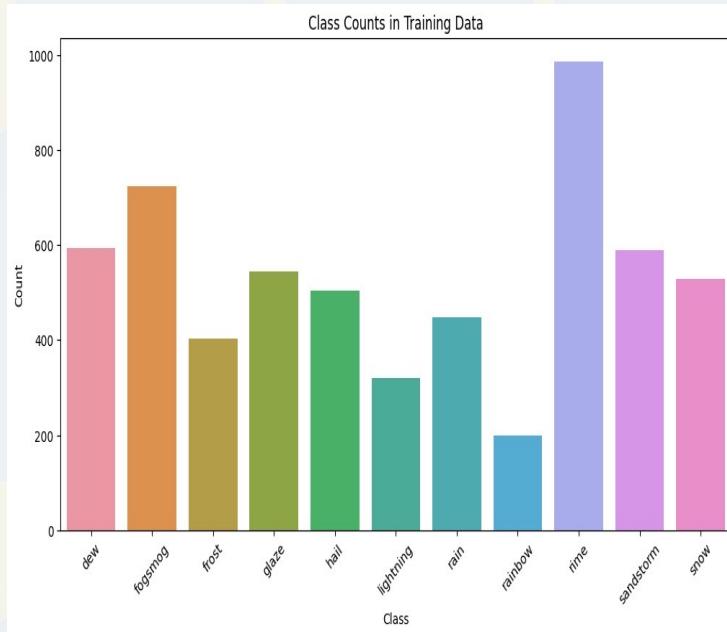
02

Approach

Sample of Data



Data Preprocessing

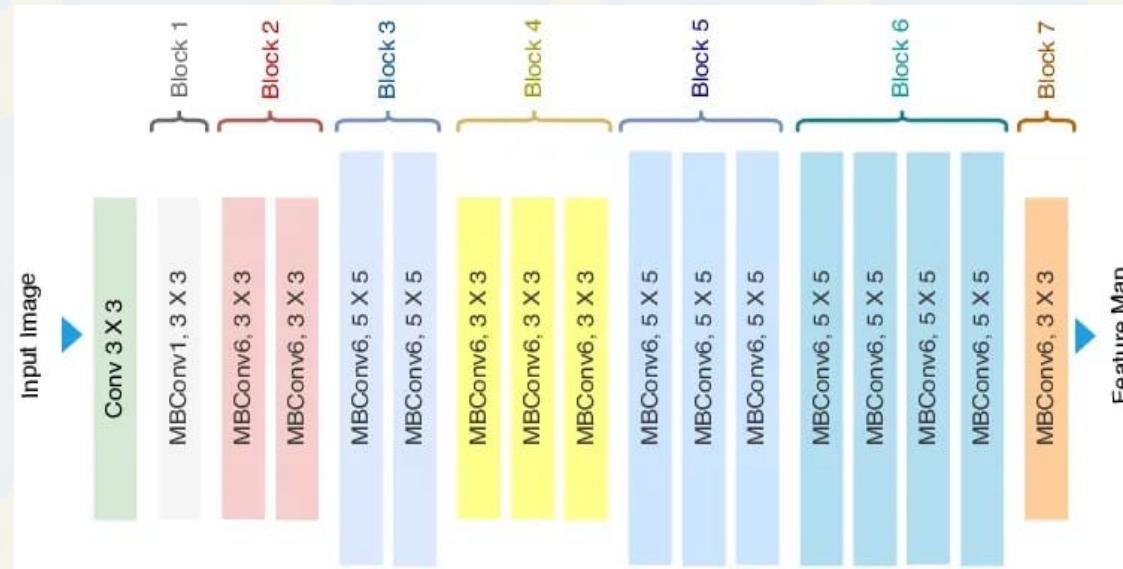


Imbalanced Data

Data Augmentation

The data is now approx.
balanced

Model Building



EfficientNet-B0
Architecture

Model Summary

- Utilizes EfficientNet-B0 pre-trained model for feature extraction in image classification tasks.
- Sequentially stacks layers including global average pooling and multiple dense layers with ReLU and LeakyReLU activations.
- Incorporates dropout layers for regularization to prevent overfitting.
- Final dense layer employs softmax activation for multi-class classification.
- Provides flexibility for model compilation, training, and potential fine-tuning of pre-trained weights.
- Offers a versatile framework for building and customizing CNNs tailored to specific image classification needs.

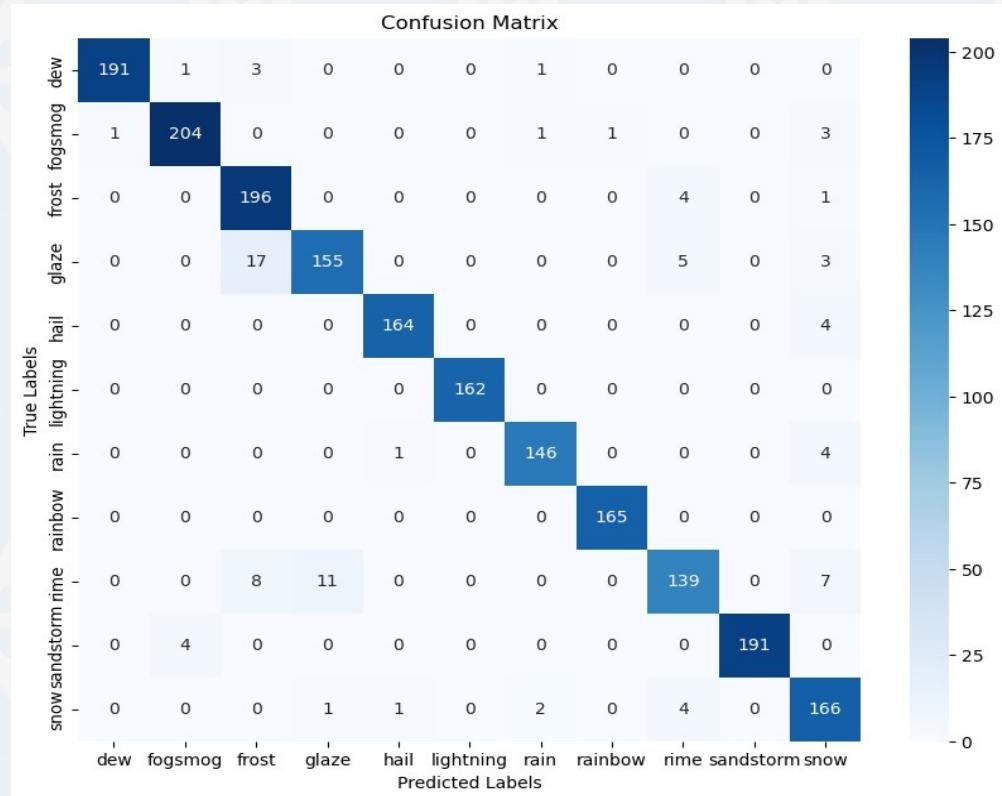


03



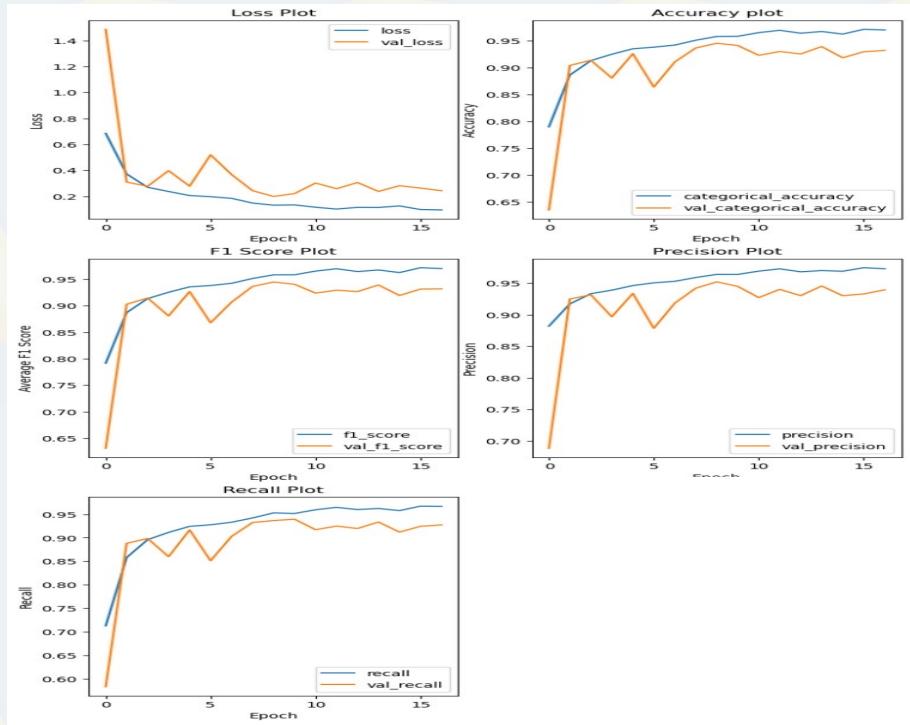
Results

Confusion Matrix



Results Curves

- We had used Accuracy(Precision, recall and F1 score) to measure how will our model is
- As we can see from the graphs there is no overfitting or underfitting.



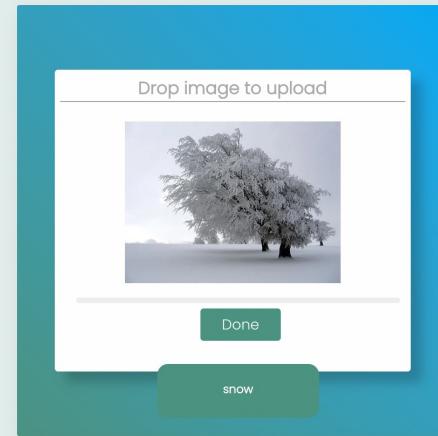
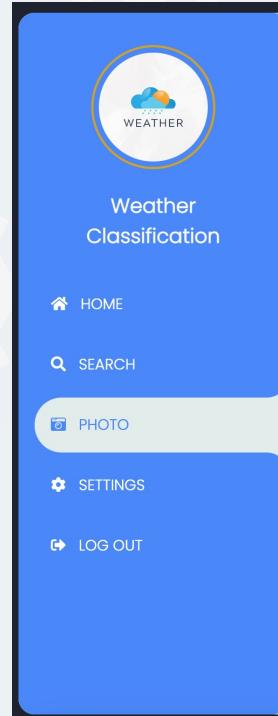
04

Deployment



Weather Classification Web

Deploy our model
using flask





Thanks!