

# Weather Effect Removal

## BY Group-30:

1. PULKIT GUPTA (18103117)
2. MANNAN BANSAL (18103121)
3. EKJOT SINGH (18103123)

## Introduction:

As the name suggests, we want to work on image restoration that has been spoiled due to bad weather. We realized that there are many instances in one's life when a good photo had been ruined due to rain or snow. With the use of AI, we want to convert this image as it has been taken on a sunny day or moonlit night. Most of the previous work generally focuses on single weather types, but we want to extend this idea to multiple kinds of weather like rain, snow, haze, fog, etc.

Demo to illustrate our project:

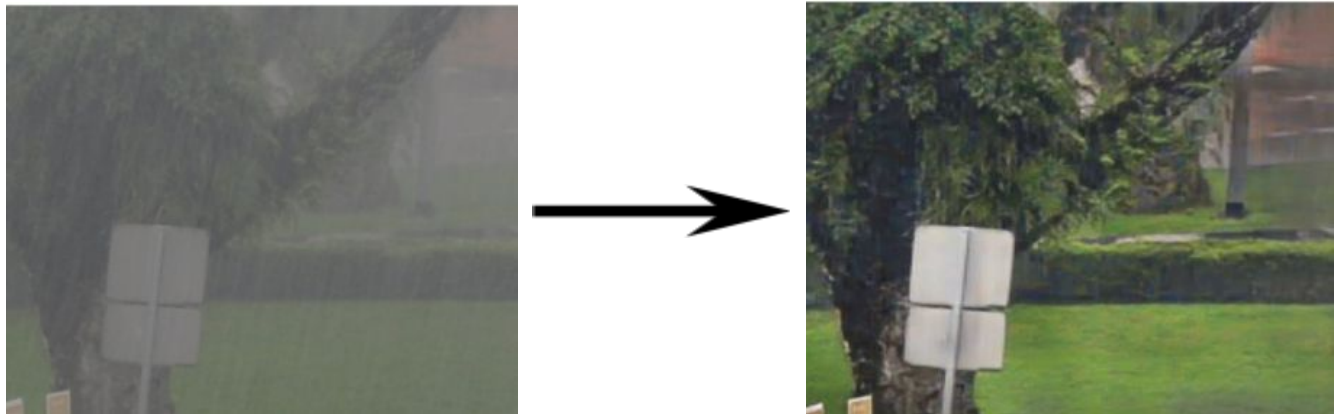


Image Source: Google Images

## Objective:

Photos serve as an integral part of someone's life. They help someone's to capture their precious moments of life. But they can be spoiled due to various means, including bad weather. So we don't want someone's memory to be spoiled. That's why anyone can use this app and get their image restored.

## **Scope:**

The app can be extended to the camera on mobiles. In this way, as soon as the user captures the image, our model should detect an image with bad weather and give the user the option to fix it at the time of saving the photo. In this way, it would be convenient for every user to use this feature.

## **Features:**

It will be open to all project, that means no headache of login or sign up.

We are planning this project with a very user-friendly interface so the user can get their image converted with a simple click of a mouse.

We will be using the high-quality dataset to train our CNN model, thus increasing our model's accuracy.

## **Related works:**

We have come across various deep learning models in image enhancement, such as super-resolution, deblurring, cartooning, image filtering, red-eye removal, etc. But there were few related to weather. These existing works have all shown the power of deep learning and how it can be explored further.

## **Requirements:**

### **1. External Interface Requirements:**

- a. Frontend: ReactJS, HTML, CSS and Javascript
- b. Backend: Python, Django
- c. Database: MySQL

### **2. Non Functional Requirements:**

- a. There is only one entity to interact with that is the User itself.
- b. The user will upload the image on our server. The image will be passed through our CNN model and after some processing, we will display the enhanced image to the user.