Water Detection

Problem Statement:

There is widespread Malaria in a city. It is affecting the school students a lot and the government noticed that it is because so many schools have water logging issues on their campuses. Nowadays every school campus has cameras installed everywhere in the campus. The school organizations want that whenever there is water logging on their school campus they should raise an alarm or notify the school administrators to solve that water logging problem. Based on this scenario, prepare a model to identify the water logging in school campuses through images.

DataSet Preparation:

As the main objection of the problem is to detect water puddles in the image, Its necessary to think about the potential water accumulation areas in the school campus or in its vicinity. Water typically gets stored in road pits , wall-floor corners or on sandy paths.
Along with the images with the water, It's also equally important to prepare the negative class elements i.e images without water logging.

Positive class images:

Wet/water surface image dataset is taken from one of the research group, Which is publicly available and can be checked from here
https://data.mendeley.com/datasets/y6zyrnxbfm/4

Negative class images:

It's a difficult task to prepare the dataset having no water content because images should resemble the positive class image without water up to certain extent.
After a lot of image survey, footpath image dataset from kaggle was selected as

- It showed similarity with positive class image
- It had potential water storage areas
- Variety of images

In order to create the robust dataset, some school campus images were taken from google photos and added to the dataset.

Note: Although the problem statement is limited to finding the water puddles in the school campus, we have used images outside of school as it is not feasible to get those exact images.

The selected dataset contains several water and non water images. It is expected that the model trained on this dataset should be able to capture all the water images.

Complete DataSet :
https://drive.google.com/drive/folders/1BOcof2DIpdNgcQy4W3hjgHVm6C20_tzM?usp=share_link

Data Processing:

All the images were resized to 224x224 pixels and normalized. 25% of data was used for testing.

Training:

Several model architectures were tested including the transfer learning ones. The one which performed relatively better is shown here:

```
CNN_model = Sequential()

   CNN_model.add(Conv2D(filters = 96, input_shape = (224, 224, 3),
kernel_size = (11, 11), strides = (4, 4), padding = 'valid'))
   CNN_model.add(Activation('relu'))
   CNN_model.add(MaxPooling2D(pool_size = (2, 2), strides = (2, 2),
padding = 'valid'))

   CNN_model.add(Conv2D(filters = 256,  kernel_size = (5, 5), strides =
(1, 1), padding = 'valid'))
   CNN_model.add(Activation('relu'))
   CNN_model.add(MaxPooling2D(pool_size = (2, 2), strides = (2, 2),
padding = 'valid'))

   CNN_model.add(Flatten())
   CNN_model.add(Dense(512))
   CNN_model.add(Activation('relu'))

   CNN_model.add(Dense(256))
   CNN_model.add(Activation('relu'))

   CNN_model.add(Dense(1, activation = 'sigmoid'))
```

Model Performance:

Evaluation criteria:

As we are dealing with the classification problem the best metrics would be one of the following scores. More precisely **Recall** because we want to minimize false negatives. I.e
There is no harm in raising an alarm if there is water accumulation but there is a problem if we don't raise the alarm if there is a water accumulation. Below you can find all the metrics.

['loss', 'accuracy']
[0.482, 0.89]

	precision	**recall**	**f1-score**	**support**
water	**0.89**	**0.91**	**0.90**	**214**
noWater	**0.91**	**0.88**	**0.89**	**206**