Machine Learning Project Report

Reza Khaleghi 2080242

Prof. p zanuttigh

Introduction

The primary goal of this project is to classify images based on weather conditions using machine learning techniques. This involves developing a model capable of accurately identifying and categorizing various weather patterns in different images.

Datasets

The project utilizes four main datasets – MWD, ACDC, UVAid, Syndrone . These datasets comprise a diverse range of images representing different weather conditions. Each dataset contains various labels like 'clear', 'fog', 'rain', etc., which are integral for the training and classification process.

The preprocessing steps involve resizing images to maintain uniformity, normalizing the image data for efficient model training, and possibly applying data augmentation techniques to enhance the diversity and robustness of the dataset.

Procedure

The project extensively uses TensorFlow, a powerful library for machine learning and neural network modeling, along with other supportive Python libraries.

The core of the project is a Convolutional Neural Network (CNN), designed with multiple layers including convolutional layers, pooling layers, and dense layers. Activation functions like ReLU and softmax are employed in different layers of the model.

The CNN model is trained using specific loss functions and optimizers. Training involves multiple epochs and batch sizes, tailored to optimize the learning process and achieve high accuracy.

Results

The model's performance is evaluated using metrics such as accuracy, precision, recall, etc. These metrics are presented in a graphical or tabular format for a clear understanding of the model's effectiveness.

An in-depth analysis of the results is carried out to understand the model's performance. This includes identifying patterns, assessing strengths and weaknesses, and addressing any challenges faced during the project.

Where applicable, the model's performance is compared with other existing models or industry benchmarks to gauge its effectiveness and efficiency.

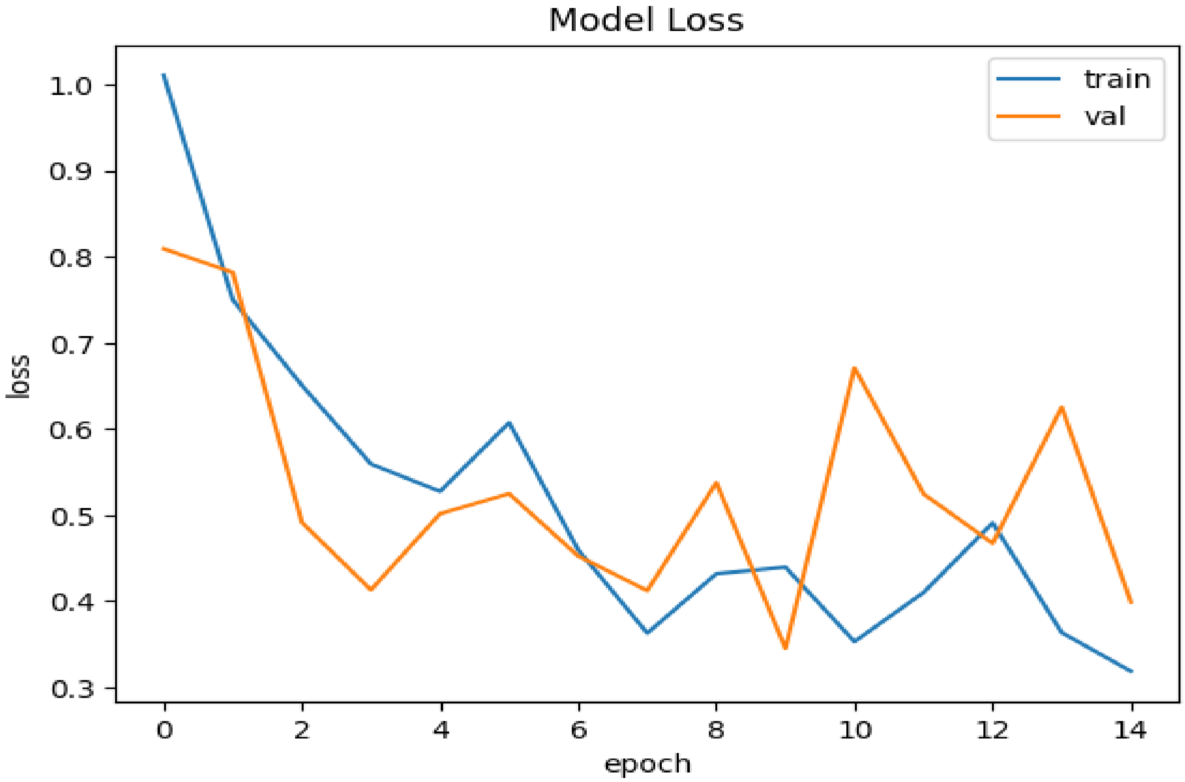
For each data set , I send details and plots and write the accuracy.

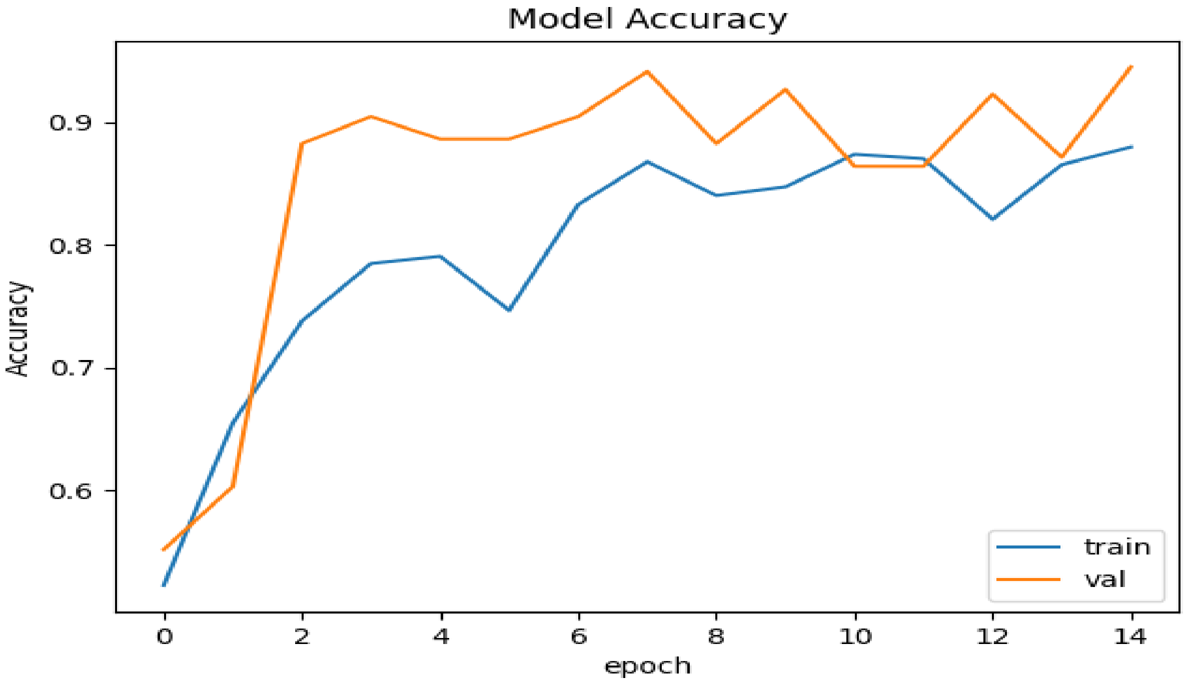
Also we use prediction for each models, by related to datasets

MWD:

```
Test Loss: 0.3909703493118286
Test Accuracy: 0.9460431933403015
```

In this model, I use 3 CNN Model, train dataset with 15 epoch and 0.001 learning rate with Adam optimizer


Model Loss
train
val
loss
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8
10
12
14
epoch


Model Accuracy
Accuracy
0.9
0.8
0.7
0.6
train
val
0
2
4
6
8
10
12
14
epoch

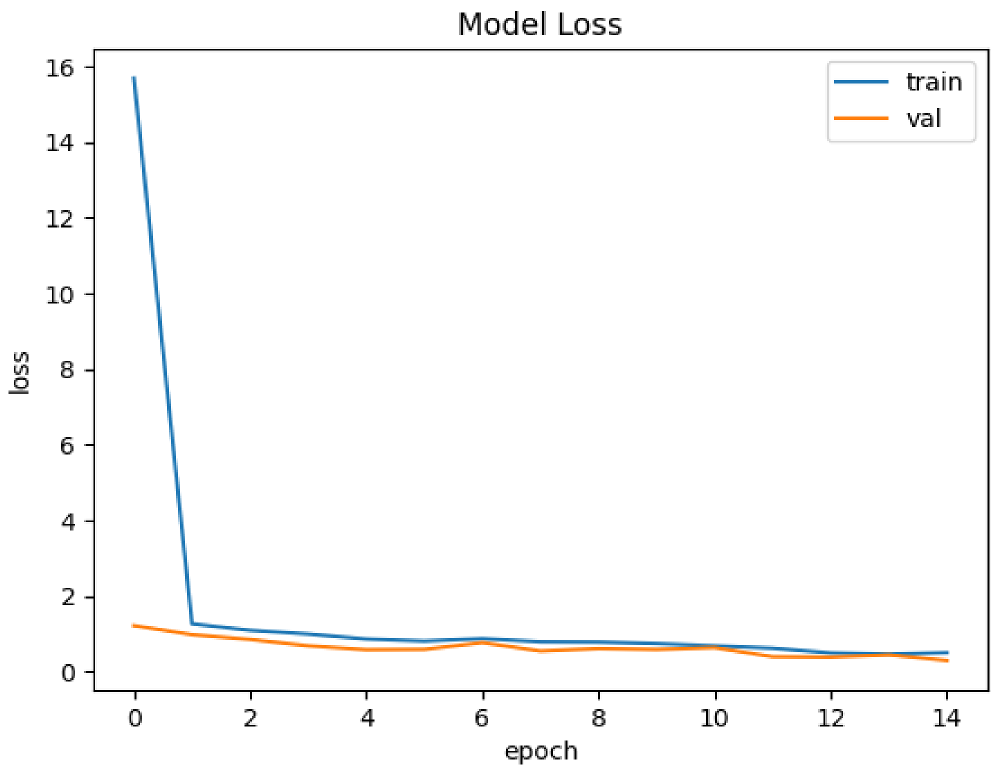
ACDC:

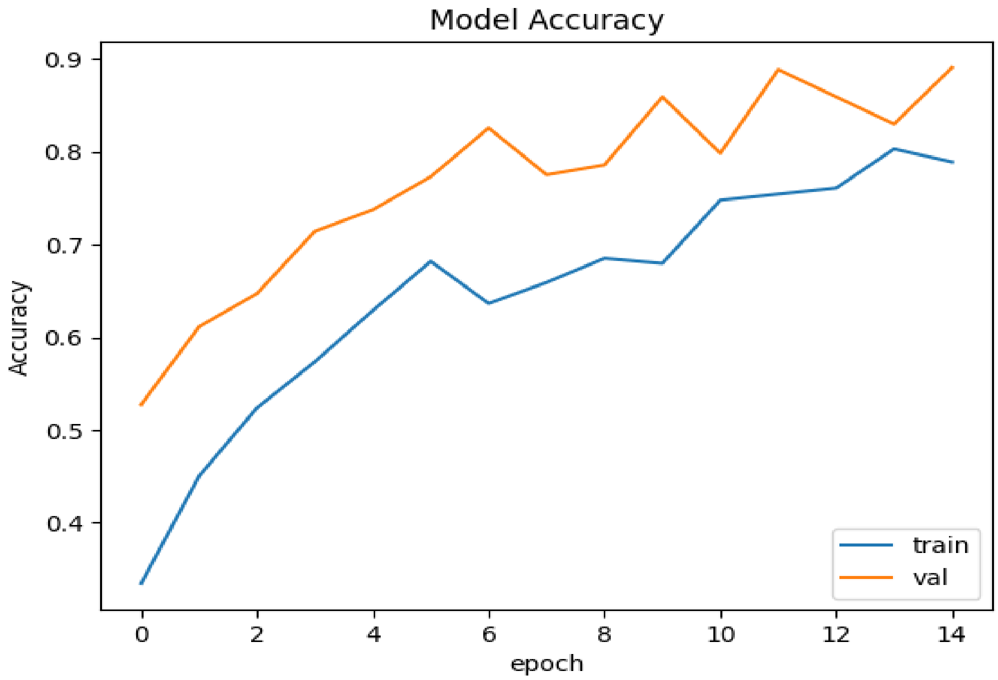
At first , I sort the dataset into 5 class ”clear , fog , rain , night ,snow” and make new datasets as in final project file says,

The badge size is 32(it can be less but because the dataset is a bit big , it’s take too time for train and give accuracy and test loss,

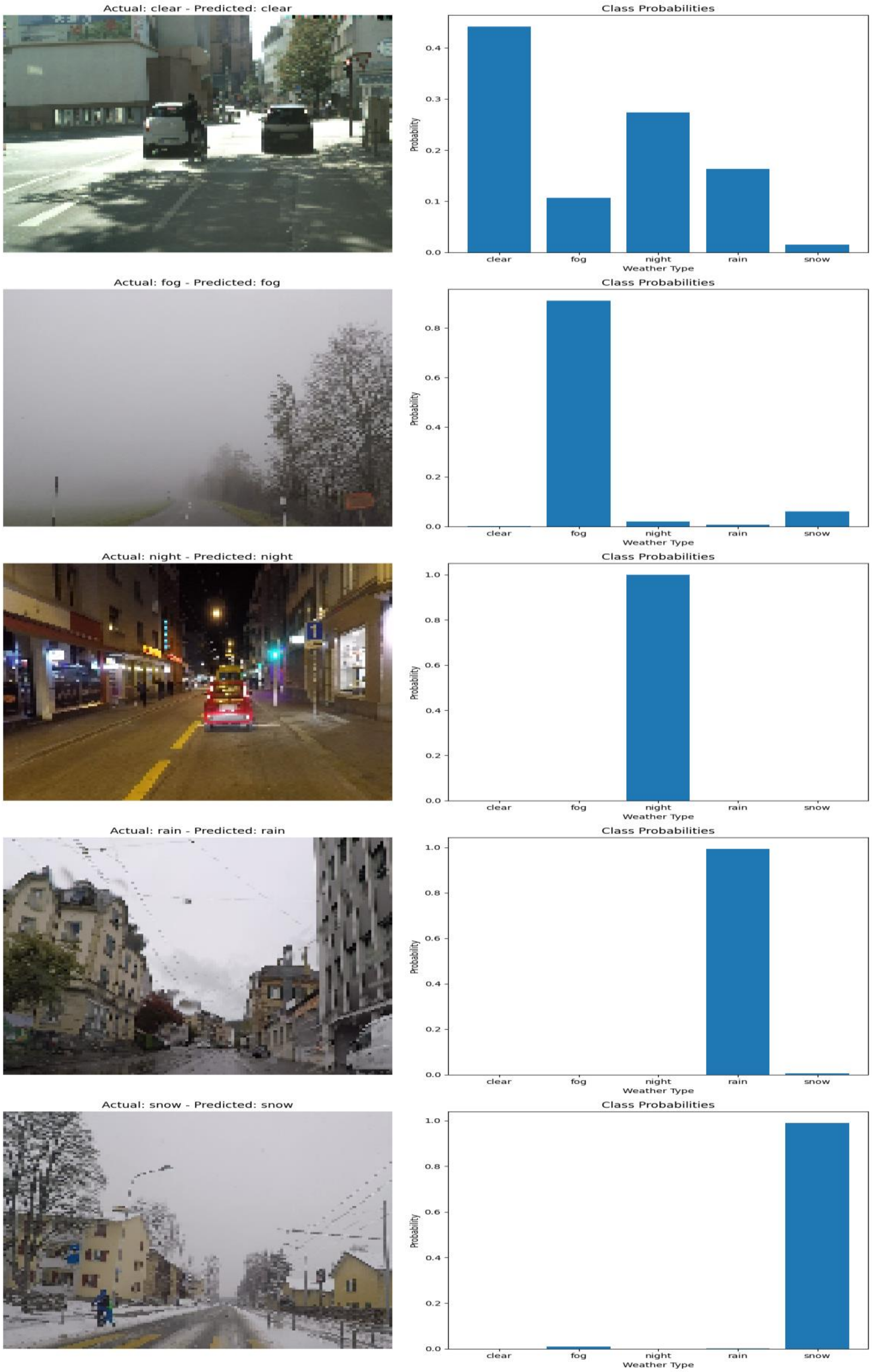
As MWD datasets , we use 3 CNN model , 15 epoch we use , for having better results

```
Test Loss: 0.29717835783958435
Test Accuracy: 0.8907563090324402
```





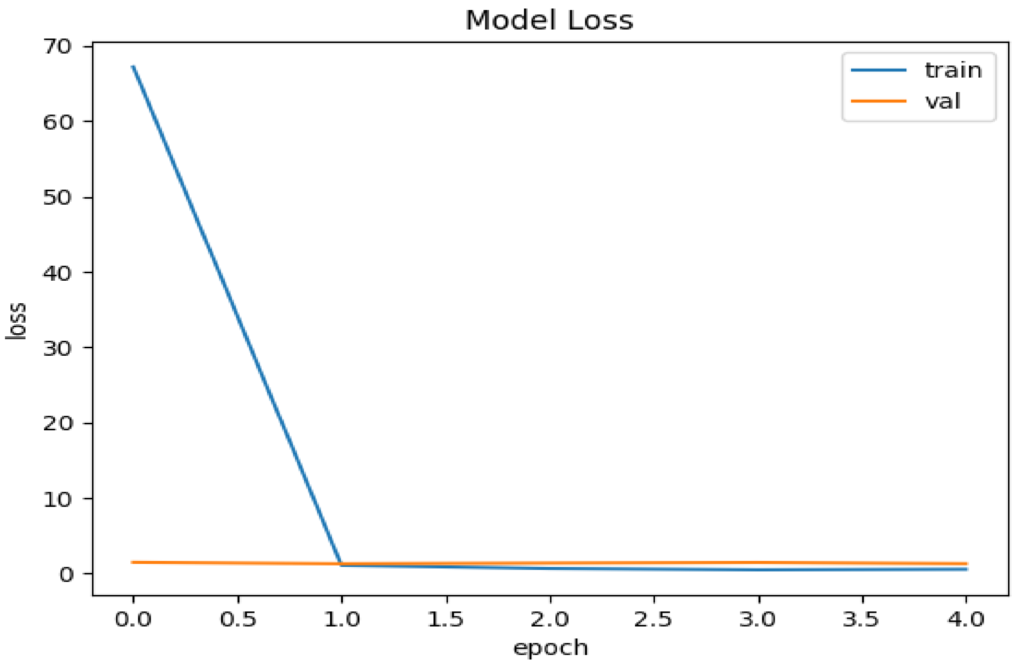
Here ,we have the prediction result of this dataset with plot

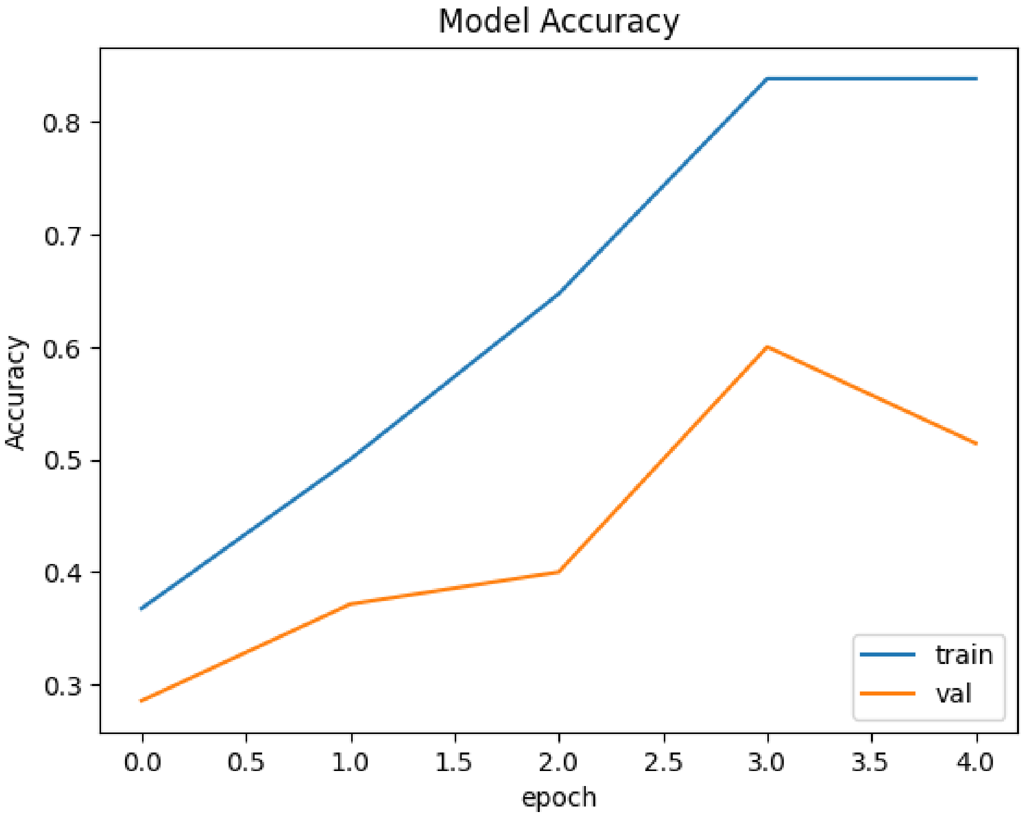

Actual: clear - Predicted: clear
Class Probabilities
Probability
0.0
0.1
0.2
0.3
0.4
clear
fog
night
rain
snow
Weather Type
Actual: fog - Predicted: fog
Class Probabilities
Probability
0.0
0.2
0.4
0.6
0.8
clear
fog
night
rain
snow
Weather Type
Actual: night - Predicted: night
Class Probabilities
Probability
0.0
0.2
0.4
0.6
0.8
1.0
clear
fog
night
rain
snow
Weather Type
Actual: rain - Predicted: rain
Class Probabilities
Probability
0.0
0.2
0.4
0.6
0.8
1.0
clear
fog
night
rain
snow
Weather Type
Actual: snow - Predicted: snow
Class Probabilities
Probability
0.0
0.2
0.4
0.6
0.8
1.0
clear
fog
night
rain
snow
Weather Type

UAVid:

In this dataset , we have 4 class

Same as ACDC , I use a function for sorting the data and classify again ,because the dataset is so small, we can’t train it as big as the others datasets ,the best result for avoid of overfitting is batch size is 1 ,we use just 2 CNN Model and train in 5 epoch


Model Loss
loss
70
60
50
40
30
20
10
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
epoch
train
val

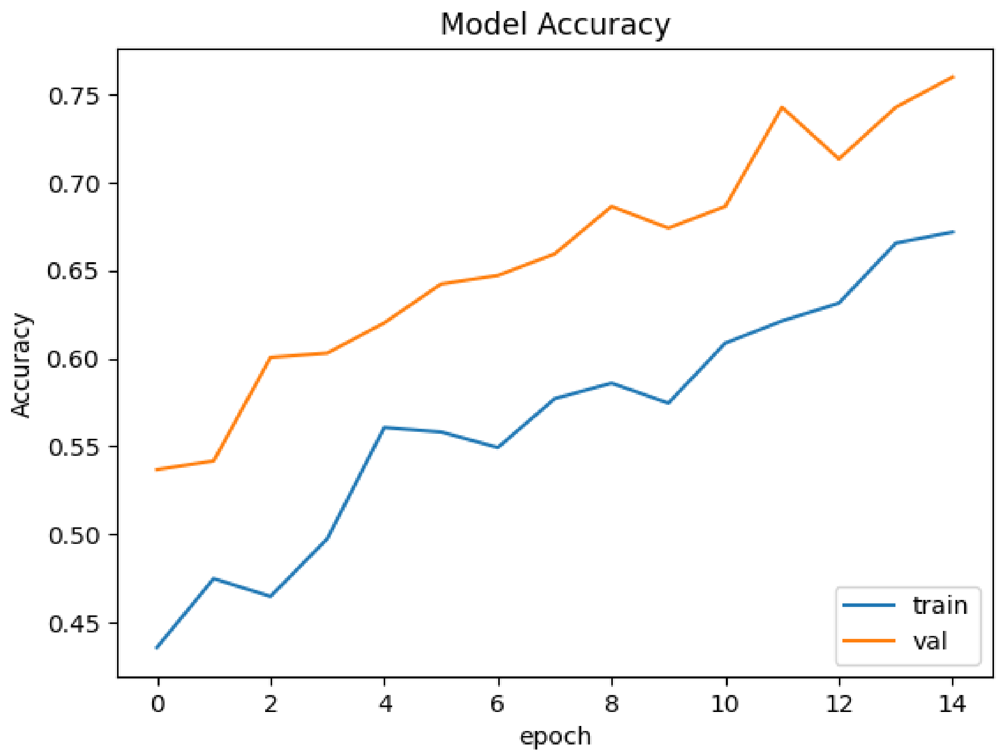

Model Accuracy
Accuracy
0.8
0.7
0.6
0.5
0.4
0.3
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
epoch
train
val

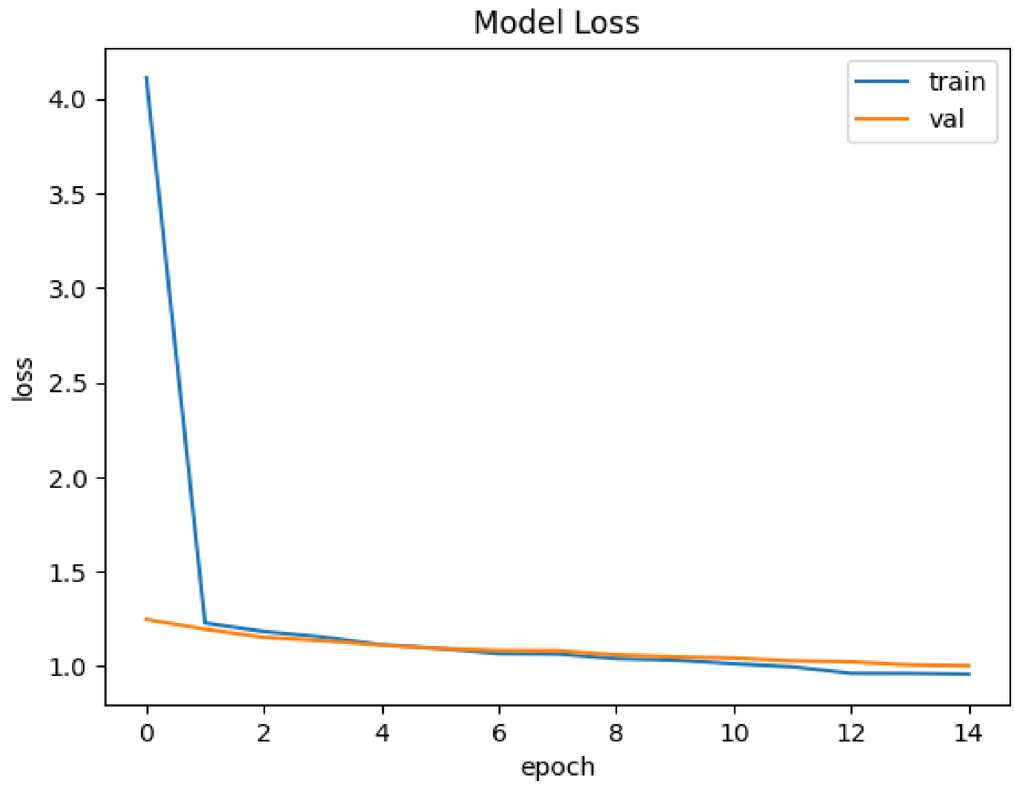
Syndrone:

This dataset is as big as ACDC but it must classify in 4 classes

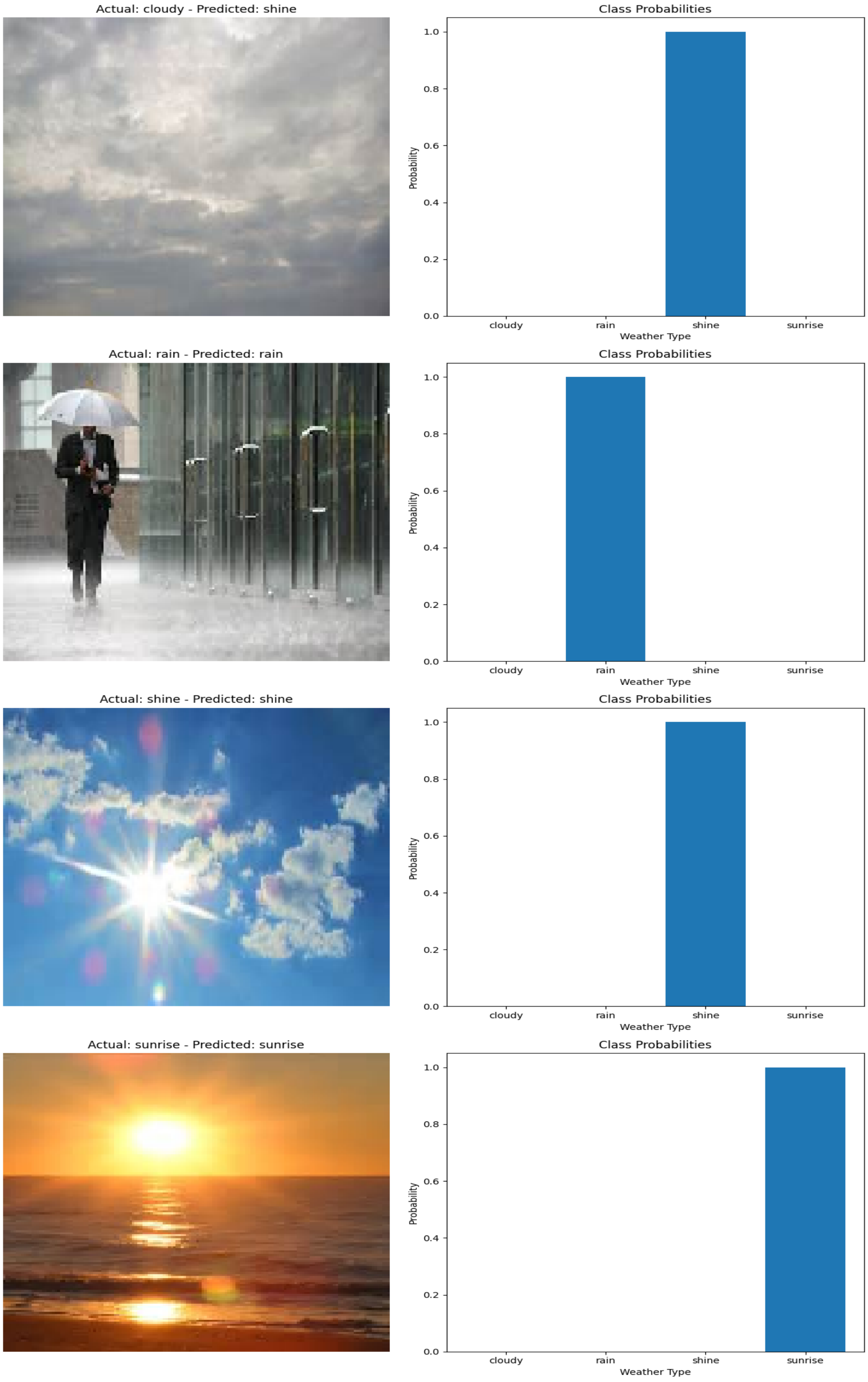
I had sort it like other dataset, the train it ,the batch size is 32 and we use 3 CNN model and 15 epoch for training datasets

```
Test Loss: 1.0017412900924683
Test Accuracy: 0.7598039507865906
```





We use the prediction function for this datasets and here the results with class probabilities


Actual: cloudy - Predicted: shine
Class Probabilities
Probability
1.0
0.8
0.6
0.4
0.2
0.0
cloudy
rain
shine
sunrise
Weather Type
Actual: rain - Predicted: rain
Class Probabilities
Probability
1.0
0.8
0.6
0.4
0.2
0.0
cloudy
rain
shine
sunrise
Weather Type
Actual: shine - Predicted: shine
Class Probabilities
Probability
1.0
0.8
0.6
0.4
0.2
0.0
cloudy
rain
shine
sunrise
Weather Type
Actual: sunrise - Predicted: sunrise
Class Probabilities
Probability
1.0
0.8
0.6
0.4
0.2
0.0
cloudy
rain
shine
sunrise
Weather Type