Artificial Intelligence and Machine Learning (6CS012)

Portfolio Project-II: Report

[Weather classification with CNN]

Student ID	: 2052267
Student Name	: Manish Darji
Group	: L6CG3
Lecturer	: Sunil Raut Kshetri
Cohort	: 5
Submission Date	: <2 /5 /2022>

Abstract

In this project we had used the CNN neural network for the classification of multi class weather image. To implement CNN, we had used the TensorFlow framework of Machine learning which include keras. To process the image of weather we had used google collab with GPU to train the CNN model. To make model we had used the multiple layers of CNN. To optimize the model, we had used the Adam optimization, and loss function of the keras Sparse Categorical Cross-entropy and train with 15 epochs which give us maximum training and validation accuracy and minimum loss function. With test data it's give high accuracy above 75% which is good for the classification of the weather image.

1.Introduction:

Data are traits or pieces of information that are gathered by observation and are generally numerical. Our data was gathered from picture sources that depict the weather, as defined by the criteria. It is the multi-class weather dataset (MWD), an important dataset used in the research article "multi-class weather identification from still image using heterogeneous ensemble approach." By extracting multiple variables for distinguishing diverse weather situations, the data provides a platform for outside weather study. (Zenghui, 2019)

Image classification is the task of assigning one (single-name arrangement) or more (multi-mark characterization) markings to a specific image (or picture acknowledgment).

1.1. Objectives and Goals:

Aim:

The goal of this project is to recognize the image and extract its characteristics.

Objective:

- To implement the visualization.
- To Implement the Conventional neural network for the weather classification.
- To implement the Different type of overfitting technique for avoid overfitting data.
- To implement data cleaning and processing
- To Analysis on the weather image

Convolutional neural networks (CNNs) are a type of multi-layer neural network. It pulls essential information from data that is organized in a grid-like pattern. The fact that you don't have to conduct any pre-processing on photos when utilizing CNNs is a major benefit. (towardsdatascience, 2021)

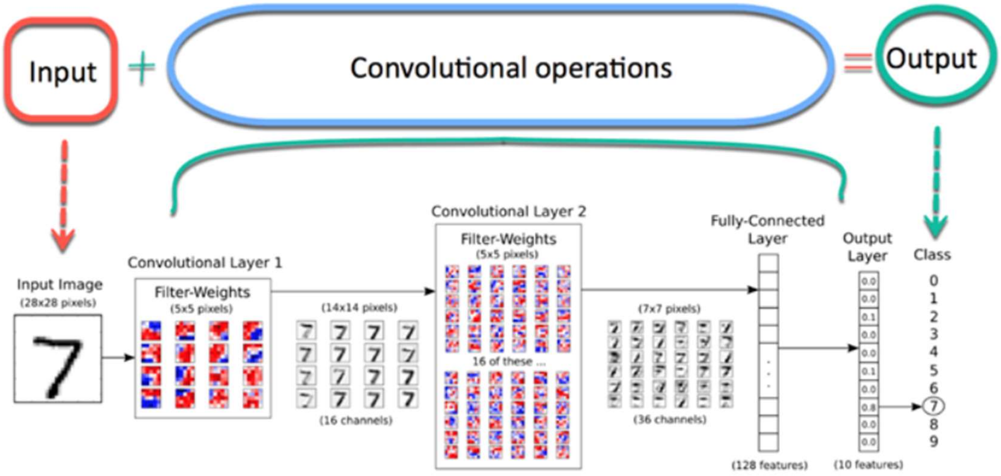


Figure 1:CNN Architecture

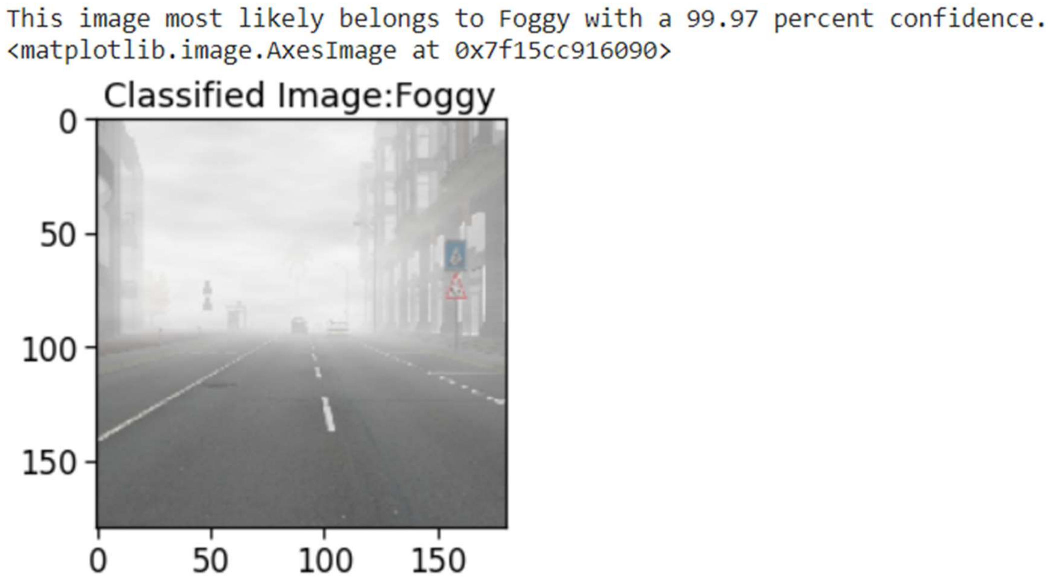
As you see in the above figure in CNN architecture the image pass to the convolution layer with different hyper parameter and different layer to make through output layer. After that it will classify the image according to the class.

1.2. Why CNN is good for image classification?

To carry out the dimensionality reduction, every of the layers of a CNN contain multiple convolutional whose filters are operating and scanning with feature matrix. This allows CNN to be a very suitable and appropriate network for any form of picture classification and processing, which is why it is useful for image classification.

2. Methodology:

In methodology we have used the CNN deep learning model to predict the image or classify the input image to the model. In the development of CNN model, we have used various type of layer and overfitting methods for CNN to make good prediction accuracy on the image. With good prediction, we can classify the input image correctly.



```
This image most likely belongs to Foggy with a 99.97 percent confidence.
<matplotlib.image.AxesImage at 0x7f15cc916090>
```

Figure 2: CNN Model predicted image

As you see in the above image is classified as the foggy which correctly defined with 99.97 accuracy with CNN model.

To make model of CNN first we import the image on the program with image processing library of TensorFlow and then divide the dataset into train and validation.

After that we make sequential model with 11 in layers. The first layer is Rescaling layer which make RGB Channel Value of [0,255] into small [0,1] range. After that we use the 2D convolution layer (Conv2D) with hyper parameters of activation with relu, kernel size. After that we used maxpoling layer with no hyperparameter and then repeat the Conv2D and Max Pooling two time. After that we use the dropout layer with this layer, which limits overfitting, changes input units to 0 at unpredictable with a repeat of rate at every movement during planning period. Inputs not set to 0 are expanded by 1/(1 - rate) with the ultimate objective that the absolute over all data sources is unaltered. After dropping out layerwe use flattern layer with no parameters. After that we used dense layer with hyper parameters of 128 unit and activation of relu. At last we use to dense layer with num_class. In my model we have 8 trainable layer which called padding, activation, relu, filter, kernel size, unit, and num_class.

We utilize a loss function called Sparse Categorical Cross Entropy with a parameter from logit to create the model. It's a loss function that determines how much cross-entropy across labels and predictions has been lost. The Adam enhance strategy, a stochastic inclination plunge technique considering versatile assessment with first and second-request minutes, is then used. Finally, there's an example of accuracy metrics. After that, we used 15 epochs to train our model.

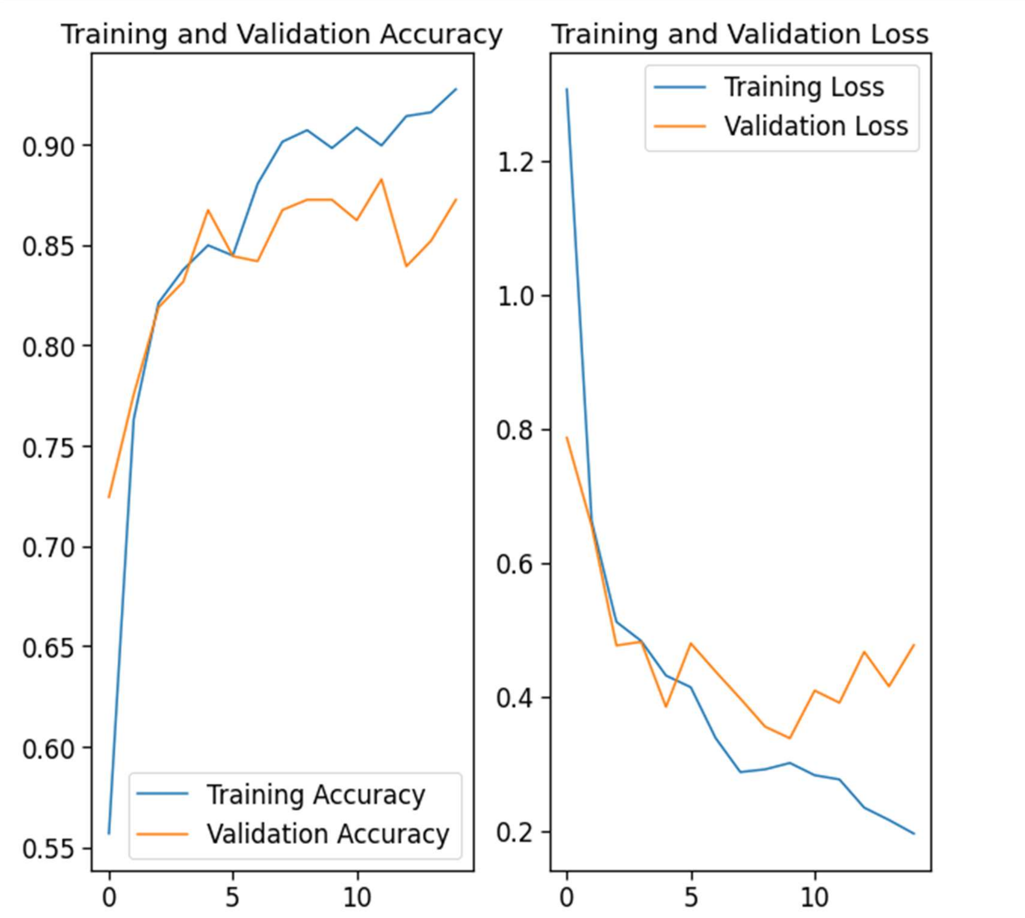
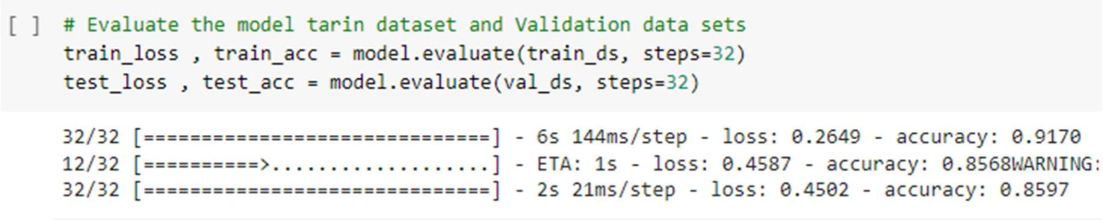


Figure 3:Training and validation loss against iteration

As you see in the above figure the training and validation accuracy has been increased with each iteration.

With each cycle, the loss in Training and Validation has minimized. We can declare that our model is good after assessing the accuracy and loss of Training and Validation. (Anon., 2022)

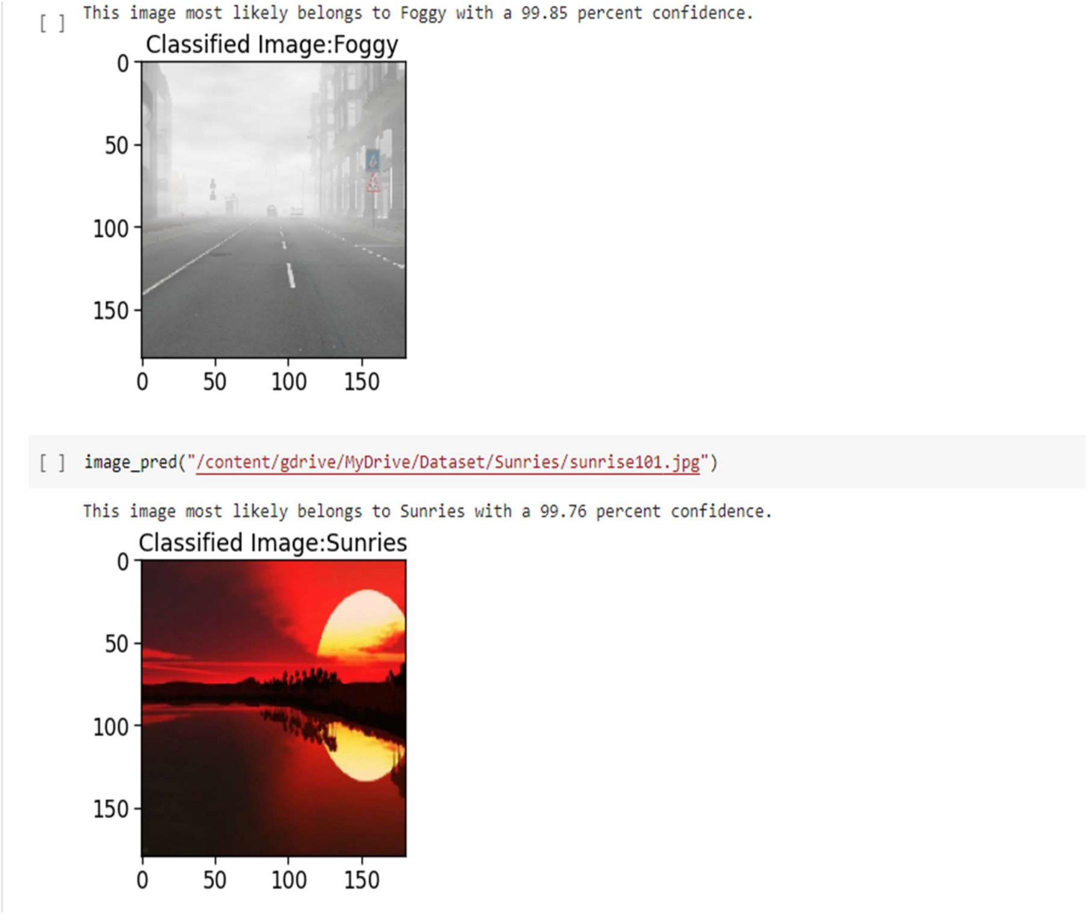
3.Evaulation of the model.



```
[ ]  # Evaluate the model tarin dataset and Validation data sets
     train_loss , train_acc = model.evaluate(train_ds, steps=32)
     test_loss , test_acc = model.evaluate(val_ds, steps=32)

     32/32 [==============================] - 6s 144ms/step - loss: 0.2649 - accuracy: 0.9170
     12/32 [==========>...................] - ETA: 1s - loss: 0.4587 - accuracy: 0.8568WARNING:
     32/32 [==============================] - 2s 21ms/step - loss: 0.4502 - accuracy: 0.8597
```

As you see in the above figure, we had evaluated the model with method called evaluate which provide the accuracy of the model that obtained from it.



```
[ ]  This image most likely belongs to Foggy with a 99.85 percent confidence.
```

```
[ ]  image_pred("/content/gdrive/MyDrive/Dataset/Sunries/sunrise101.jpg")

     This image most likely belongs to Sunries with a 99.76 percent confidence.
```

Figure 4: Test image of the model,

As you see in the above figure our model has classified correctly. With good accuracy from this point, we can say that our CNN model is good for the weather classification.

4.Conclusion

In the end this project gives lots of knowledge about the deep learning which help me for the further study in Machine learning study. With this project I can know the CNN network of the deep learning and how it's work and why it's important on the computer vision. From that knowledge it will help me to better understand on the deep learning.

Bibliography

Anon., 2022. *Image classification.* [Online]
Available at: https://www.tensorflow.org/tutorials/images/classification
[Accessed 1 5 2022].

towardsdatascience, 2021. *Understanding CNN (Convolutional Neural Network).* [Online]
Available at: https://towardsdatascience.com/understanding-cnn-convolutional-neural-network-69fd626ee7d4
[Accessed 1 5 2022].

Zenghui, W., 2019. MULTI-CLASS WEATHER CLASSIFICATION. *2019 SAUPEC/RobMech/PRASA Conference,* 1(1), pp. 28-30.