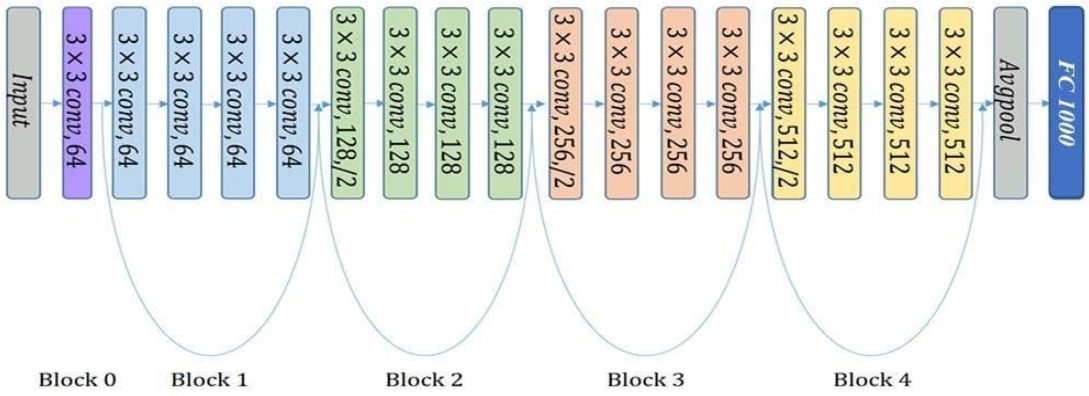
Project Presentation

- Architecture Used In the paper:

  Res-Net 18 .

  Normally, ResNet composed of a number of modules with four convolutional layers in each module. By configuring different numbers of channels and residual blocks in the module, we can create different ResNet models, such as ResNet-50 (has 50-layers), ResNet-152 (has 152 layers), or ResNet-18 (has 18 layers). Fig.2 shows the structure of ResNet-18. Together with the first convolutional layer and the final fully connected layer, there are 18 layers in total. ResNet 18 contains five blocks (modules): the 0 th block is one single 3×3 convolutional layer, and each of the rest contains four 3×3 convolutional layers [27] .



- Dataset Details:

Dataset name: Weather Classification

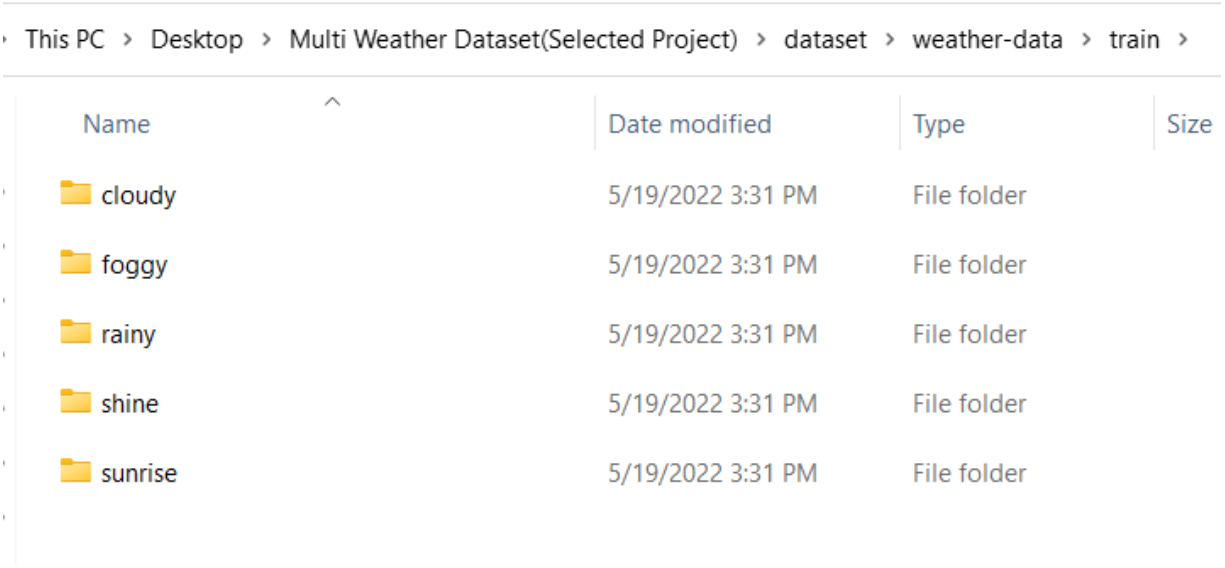
Link: https://www.kaggle.com/code/kamalkhumar/weather-classification-with-augmentation/data

Total Number Of Samples in the dataset: 1500

The Dimension of images: (224,224,3)

Number of Classes:

There are 5 classes ['cloudy', 'foggy', 'rainy', 'shine', 'sunrise']



- Implementation Details:

The dataset is divided into 75% train and 25% validation(Testing).

Train Data:

```
Training cloudy images are: 225
Training foggy images are: 225
Training rainy images are: 225
Training shine images are: 187
Training sunrise images are: 262
Total: 1124
```

Validation(testing data):

```
Valid cloudy images are: 75
Valid foggy images are: 75
Valid rainy images are: 75
Valid shine images are: 63
Valid sunrise images are: 88
Total:376
```

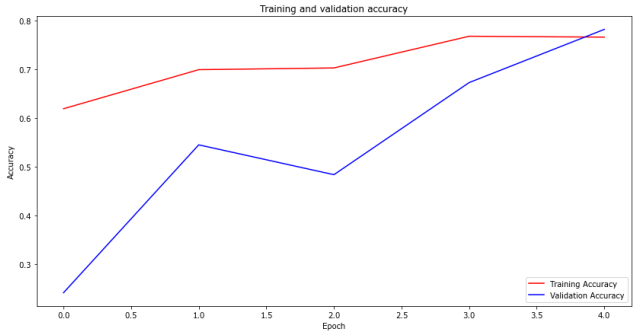
Hyperparameters: stochastic Gradient Descent with momentum and decay

Optimizers: Adam

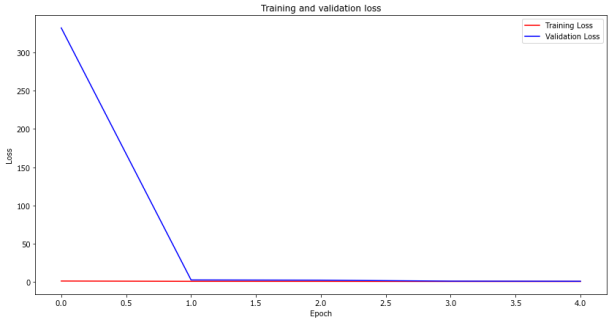
```
Model: "res_net18_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_100 (Conv2D)          multiple                  9472
_________________________________________________________________
batch_normalization_100 (Bat multiple                  256
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 multiple                  0
_________________________________________________________________
resnet_block_40 (ResnetBlock multiple                  74368
_________________________________________________________________
resnet_block_41 (ResnetBlock multiple                  74368
_________________________________________________________________
resnet_block_42 (ResnetBlock multiple                  231296
_________________________________________________________________
resnet_block_43 (ResnetBlock multiple                  296192
_________________________________________________________________
resnet_block_44 (ResnetBlock multiple                  921344
_________________________________________________________________
resnet_block_45 (ResnetBlock multiple                  1182208
_________________________________________________________________
resnet_block_46 (ResnetBlock multiple                  3677696
_________________________________________________________________
resnet_block_47 (ResnetBlock multiple                  4723712
_________________________________________________________________
global_average_pooling2d_5 ( multiple                  0
_________________________________________________________________
flatten_5 (Flatten)          multiple                  0
_________________________________________________________________
dense_5 (Dense)              multiple                  2565
=================================================================
Total params: 11,193,477
Trainable params: 11,183,877
Non-trainable params: 9,600
```

- Results and Visualizations:

Accuracy Learning Curve:



Loss Learning Curve:



Testing Accuracy:

```
In [91]: ModelLoss, ModelAccuracy = model.evaluate(validation_generator)

         print('Model Loss is {}'.format(ModelLoss))
         print('Model Accuracy is {}'.format(ModelAccuracy))

         24/24 [==============================] - 12s 492ms/step - loss: 0.8934 - accuracy: 0.7819
         Model Loss is 0.8934231400489807
         Model Accuracy is 0.7819148898124695
```

Confusion Matrix: