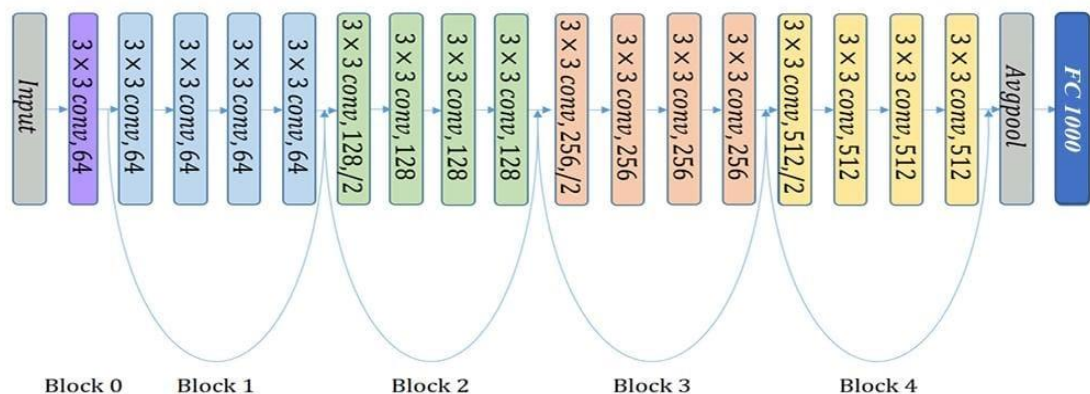


Project Presentation

- 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. ResNet18 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: (256,256,3)

Number of Classes:

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

This PC > Desktop > Multi Weather Dataset(Selected Project) > dataset > weather-data > train >				
Name		Date modified	Type	Size
cloudy		5/19/2022 3:31 PM	File folder	
foggy		5/19/2022 3:31 PM	File folder	
rainy		5/19/2022 3:31 PM	File folder	
shine		5/19/2022 3:31 PM	File folder	
sunrise		5/19/2022 3:31 PM	File folder	

- 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

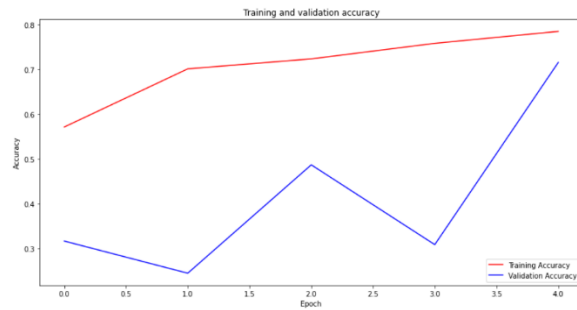
Optimizers: Adam

Model: "res_net18_5"

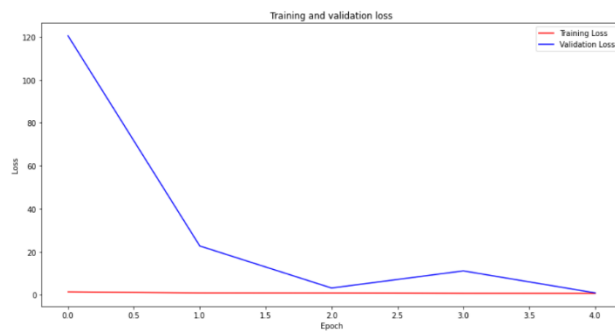
Layer (type)	Output Shape	Param #
conv2d_100 (Conv2D)	multiple	9472
batch_normalization_100 (Batch Normalization)	multiple	256
max_pooling2d_5 (MaxPooling2D)	multiple	0
resnet_block_40 (ResnetBlock)	multiple	74368
resnet_block_41 (ResnetBlock)	multiple	74368
resnet_block_42 (ResnetBlock)	multiple	231296
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 (Global Average Pooling2D)	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:

```
5]: ModelLoss, ModelAccuracy = model.evaluate(validation_generator)
print('Model Loss is {}'.format(ModelLoss))
print('Model Accuracy is {}'.format(ModelAccuracy))
```

24/24 [=====] - 3s 131ms/step - loss: 0.8675 - accuracy: 0.7154
 Model Loss is 0.8675402402877808
 Model Accuracy is 0.7154255509376526

Confusion Matrix:

