Chapter 1

INTRODUCTION

Weather plays a great role in human life. Many of our daily activities and businesses depend on weather conditions. As well as, there is a huge life and property loss due to unexpected Weather conditions. If we are able to efficiently predict the weather conditions for the future, then we can prevent or minimize these losses. The atmospheric condition of the earth does not remain same at every time. We have various seasons like summer, winter, spring, autumn, Monsoon, etc. Weather changes from time to time. This weather change is rather normal and regular phenomenon of the world. In many outdoor applications for computer vision, the “bad” weather situations, such as haze, fog, rain, hail and snow, are involved. And it is urgent to detect and recognize the various outdoor weather situations, especially the severe ones. Meanwhile, the observation of weather situations in meteorology is still mainly rely on manual, and weather situation is not exactly the same even within every small region. Therefore, automatic recognition of the outdoor weather situation based on image or video data gets more extensive attention in recent years.

Chapter 2

LITERATURE SURVEY

Kurihata et al. [1,2] Many vehicles assistant driving systems use weather recognition to improve the road safety. For example, they can set speed limit in extreme weather conditions, automatically open the wiper in a rainy day and so forth. Hand-crafted features are popular in these works. proposed that rain drops are strong cues for the 100 presence of rainy weather and developed a rain feature to detect rain drops on the windshield.

Pavli et al. [3] transformed images into frequency domain and detected the presence of fog through training different scaled and oriented Gabor filters in the power spectrum. Although the aforementioned approaches have shown good performance, they are usually limited to the in-vehicle perspective and cannot be applied to wider range of applications.

Zhao et al. [4] pointed out that pixel-wise intensities of dynamic weather conditions (rainy, snowy, etc.) fluctuate over time while static weather conditions (sunny, foggy, etc.) almost stay unchanged. They proposed a two120 stage classification scheme which first distinguishes between the two conditions then utilizes several spatio-temporal and chromatic features to further estimate the weather category.

A. G. Salman, B. Kanigoro and Y. Heryadi [5] Weather forecasting has gained attention many researchers from various research communities due to its effect to the global human life. The emerging deep learning techniques in the last decade coupled with the wide availability of massive weather observation data and the advent of information and computer technology have motivated many researches to explore hidden hierarchical pattern in the large volume of weather dataset for weather forecasting. This study investigates deep learning techniques for weather forecasting. In particular, this study will compare prediction performance of Recurrence Neural Network (RNN), Conditional Restricted Boltzmann Machine (CRBM), and Convolutional Network (CN) models. Those models are tested using weather dataset provided by BMKG (Indonesian Agency for Meteorology, Climatology, and

Geophysics) which are collected from a number of weather stations in Aceh area from 1973 to 2009 and El-Nino Southern Oscilation (ENSO) data set provided by International Institution such as National Weather Service Center for Environmental Prediction Climate (NOAA). Forecasting accuracy of each model is evaluated using Frobenius norm. The result of this study expected to contribute to weather forecasting for wide application domains including flight navigation to agriculture and tourism.

G. Howard, M. Zhu, B. Chen, [6] Depth wise separable convolution is a combination of depth wise and pointwise convolution. The former separately applies a filter to each input channel, and the latter then applies a 1×1 convolution to combine the outputs of the depth wise convolution Generally, the computational cost of depth wise separable convolution is 8 to 9 times less than the standard convolution but with reduction in accuracy.

Guofa Li [7] — Effectively detecting pedestrians in various environments would significantly improve driving safety for autonomous vehicles. However, the degraded visibility and blurred outline and appearance of pedestrian images captured during hazy weather strongly limit the effectiveness of current pedestrian detection methods. To solve this problem, this paper presents three novel deep learning approaches based on Yolo.

The depth wise separable convolution and linear bottleneck skills were used to reduce the computational cost and number of parameters, rendering our network more efficient. We also innovatively developed a weighted combination layer in one of the approaches by combining multi-scale feature maps and a squeeze and excitation block. Collected pedestrian images in hazy weather were augmented using six strategies to enrich the database. Experimental results show that our proposed methods can effectively detect pedestrians in hazy weather, significantly outperforming state-of-the-art methods in both accuracy and speed.

Li et al. [8] presented a benchmark including both synthetic and real-world rainy images with some rain types to investigate draining algorithms in traffic monitoring scene and vehicle detection. However, these methods are mainly evaluated on rendered synthetic fog/rain images and few real images assuming specific fog/rain

models. It is thus unclear how these algorithms would be proceeding on various adverse weather conditions and how the progress could be measured in the wild.

Sakaridis et al. [9] proposed a convolution neural network (CNN) based model to generate synthetic fog on real vehicle images to investigate defogging algorithms in the traffic environments.

Hodges et al. [10] manipulated dehazing models by a dehazing network to reform full images and a discriminator network to fine tuning the enhancement weights parameters to increase the performance of vehicle detection on a dataset of synthetic foggy/hazy images.

Chapter 3

PROBLEM DEFINITION

This project shows weather recognition using dataset with eleven classes: dew, fog smoke, frost, glaze, hail, lightening, rain, rainbow, rime, sandstorm, snow. In this project 9 deep learning models are trained and checked for best accuracy out of EfficientNetB7, ResNet, Mobile Net, VGG19, Xception, InceptionResNetV2, VGG16, ResNet101, DenseNet201.

Chapter 4

METHODOLOGY

Weather Recognition plays an important role in our daily lives and many applications. The transmission of weather information of a location at certain time intervals affects the living conditions of the people there directly or indirectly. However, the recent mention of the name of artificial intelligence technology in every field has made it compulsory for computer systems to benefit from this technology. The dataset used in the study has eleven classes: dew, fog smoke, frost, glaze, hail, lightening, rain, rainbow, rime, sandstorm, snow. In the study 9 deep learning models are used namely EfficientNetB7, ResNet, Mobile Net, VGG19, Xception, InceptionResNetV2, VGG16, ResNet101, DenseNet201.

Figure 1: Dataset (dew, fog smoke, frost, glaze, hail, lightening, rain, rainbow, rime, sandstorm, snow).

3.1 Data Visualization and Exploratory Data Analysis (EDA):

Exploratory data analysis is a simple classification technique usually done by visual methods. It is an approach to analyzing data sets to summarize their main characteristics. When you are trying to build a machine learning model you need to be pretty sure whether your data is making sense or not.

The need of EDA: Every deep learning problem solving starts with EDA. It is probably one of the most important part of a deep learning project. With the growing market, the size of data is also growing. It becomes harder for companies to make decision without proper analyzing it. With the use of charts and certain graphs, one can make sense out of the data and check whether there is any relationship or not. Various plots are used to determine any conclusions. This helps the company to make a firm and profitable decisions. Once **Exploratory Data Analysis** is complete and insights are drawn, its feature can be used for supervised and unsupervised machine learning modelling.

Data Visualization:

Data Visualization represents the text or numerical data in a visual format, which makes it easy to grasp the information the data express. We, humans, remember the pictures more easily than readable text, so Python provides us various libraries for data visualization like matplotlib, seaborn, plotly, etc. In this tutorial, we will use Matplotlib and seaborn for performing various techniques to explore data using various plots.

3.2 Training Models:

a.EfficientNetB-7:

Efficient Net, first introduced in Tan and Le, 2019 is among the most efficient models (i.e. requiring least FLOPS for inference) that reaches State-of-the-Art accuracy on both ImageNet and common image classification transfer learning tasks.

The smallest base model is similar to MnasNet, which reached near-SOTA with a significantly smaller model. By introducing a heuristic way to scale the model, Efficient Net provides a family of models (B0 to B7) that represents a good combination of efficiency and accuracy on a variety of scales. Such a scaling heuristics (compound-scaling, details see Tan and Le, 2019) allows the efficiency-oriented base model (B0)

to surpass models at every scale, while avoiding extensive grid-search of hyper parameters.

People may have the impression that Efficient Net is a continuous family of models created by arbitrarily choosing scaling factor. However, choice of resolution, depth and width are also restricted by many factors:

- Resolution: Resolutions not divisible by 8, 16, etc. cause zero-padding near boundaries of some layers which wastes computational resources. This especially applies to smaller variants of the model, hence the input resolution for B0 and B1 are chosen as 224 and 240.

- Depth and width: The building blocks of Efficient Net demands channel size to be multiples of 8.

- Resource limit: Memory limitation may bottleneck resolution when depth and width can still increase. In such a situation, increasing depth and/or width but keep resolution can still improve performance.

As a result, the depth, width and resolution of each variant of the EfficientNet models are hand-picked and proven to produce good results, though they may be significantly off from the compound scaling formula. Therefore, the keras implementation (detailed below) only provide these 8 models, B0 to B7, instead of allowing arbitray choice of width / depth / resolution parameters.

b.ResNet:

After the celebrated victory of AlexNet at the LSVRC2012 classification contest, deep Residual Network was arguably the most groundbreaking work in the computer vision/deep learning community in the last few years. ResNet makes it possible to train up to hundreds or even thousands of layers and still achieves compelling performance**.** Deep networks are hard to train because of vanishing gradient problem while updating. Taking advantage of its powerful representational ability, the performance of many computer vision applications other than image classification have been boosted, such as object detection and face recognition.

Since ResNet blew people's mind in 2015, many in the research community have dived into the secrets of its success, many refinements have been made in the architecture. This article is divided into two parts, in the first part I am going to give a little bit of background knowledge for those who are unfamiliar with ResNet, in the second I will review some of the papers I read recently regarding different variants and interpretations of the ResNet architecture.

c.MobileNet:

MobileNet is a type of convolutional network designed for mobile and embedded vision applicatios. They are based on a streamlined architecture that uses depthwise separable convolutions to build lightweight deep neural networks that can have low latency for mobile and embedded devices. class of efficient models called MobileNets for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks. We introduce two simple global hyper-parameters that efficiently trade off between latency and accuracy. These hyper-parameters allow the model builder to choose the right sized model for their application based on the constraints of the problem. We present extensive experiments on resource and accuracy tradeoffs and show strong performance compared to other popular models on ImageNet classification. We then demonstrate the effectiveness of MobileNets across a wide range of applications and use cases including object detection, finegrain classification, face attributes and large scale geo-localization.

d.VGG19 AND VGG16

VGG-19 is a convolutional neural network that is 19 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object

categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

VGG16 is a convolution neural net (CNN) architecture which was used to win ILSVR(Imagenet) competition in 2014. It is considered to be one of the excellent vision model architecture till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameters they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC (fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx) parameters.

Compared with VGG16, VGG19 is slightly better but requests more memory. VGG16 model is composed of convolutions layers, max pooling layers, and fully connected layers. The total is 16 layers with 5 blocks and each block with a max pooling layer.

e.Xception:

Inspired by Google's Inception model, Xception is based on an 'extreme' interpretation of the Inception model. The Xception architecture is a linear stack of depthwise separable convolution layers with residual connections. Simple and modular architecture

e.Inception-ResNet-v2

Inception-ResNet-v2 is a convolutional neural architecture that builds on the Inception family of architectures but incorporates residual connections (replacing the filter concatenation stage of the Inception architecture).

f.ResNet101

ResNet-101 is a convolutional neural network that is 101 layers deep. You can load a pre-trained version of the network trained on more than a million images from the ImageNet database. The pre-trained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

g.DenseNet-201

DenseNet-201 is a convolutional neural network that is 201 layers deep. You can load a pre-trained version of the network trained on more than a million images from the ImageNet database. The pre-trained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

Chapter 5

RESULTS AND DISCUSSION

The dataset used in the study has eleven classes: dew, fog smoke, frost, glaze, hail, lightening, rain, rainbow, rime, sandstorm, snow.

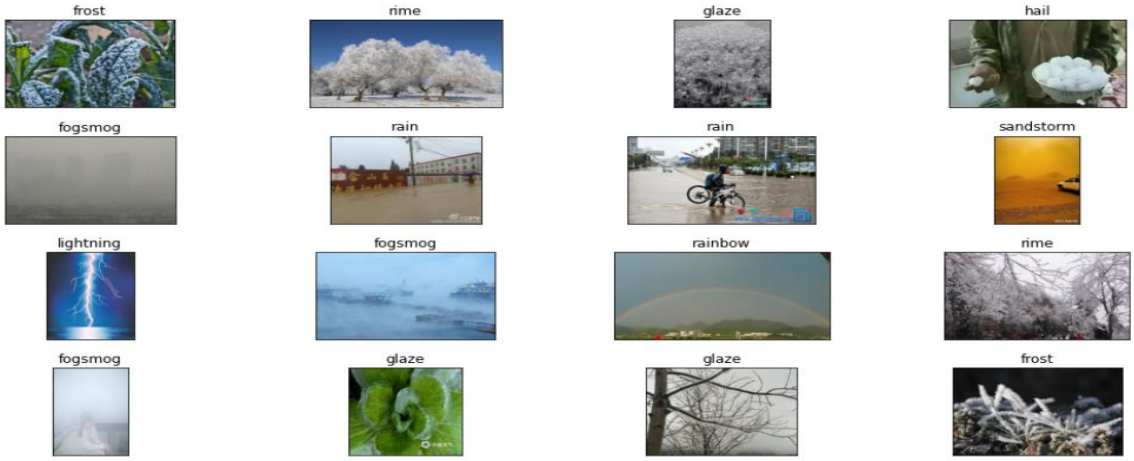


Figure 2: Weather types.

Bar graph of label count:

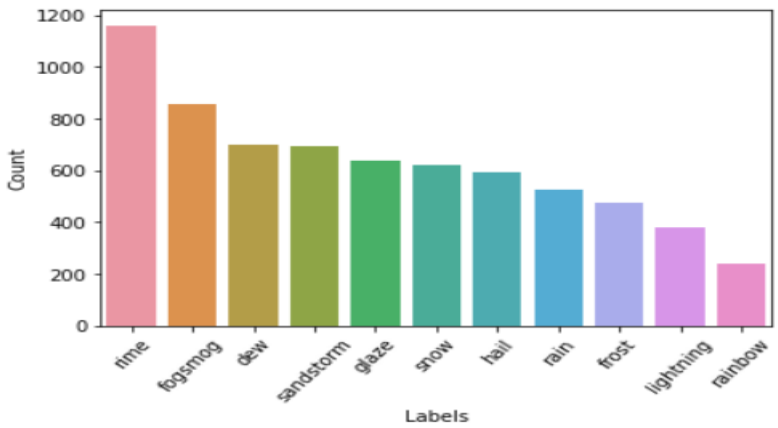


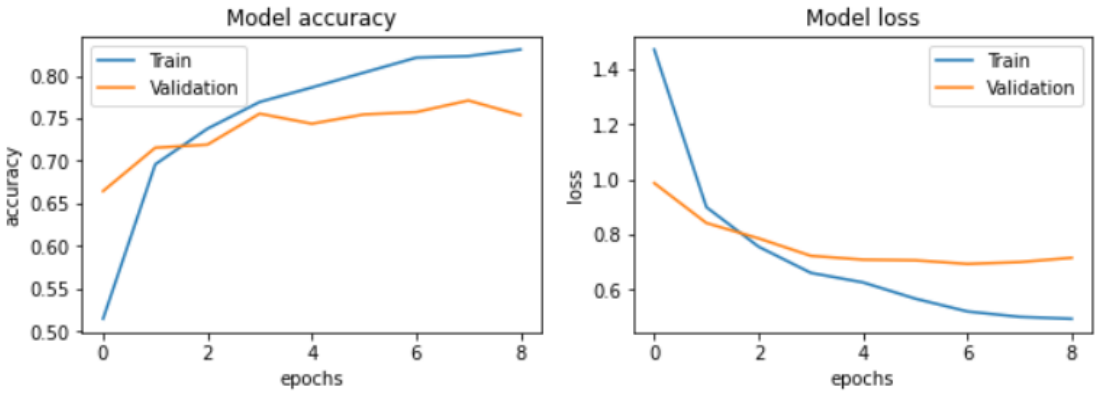
Figure 3: Bar graph.

Results of training models:

EfficientNetB-7:

The values of precision, recall, f1-score, and support. The accuracy of EfficientNetB-7 is 76.82 % and loss is 0.69611.

	precision	recall	f1-score	support
dew	0.87	0.82	0.85	146
fogsmog	0.62	0.94	0.75	173
frost	0.62	0.57	0.59	93
glaze	0.65	0.73	0.69	147
hail	0.80	0.91	0.85	102
lightning	0.95	0.77	0.85	73
rain	0.84	0.64	0.72	107
rainbow	0.98	0.93	0.95	44
rime	0.88	0.81	0.84	245
sandstorm	0.78	0.72	0.75	126
snow	0.78	0.56	0.65	120
accuracy			0.77	1376
macro avg	0.80	0.76	0.77	1376
weighted avg	0.78	0.77	0.77	1376



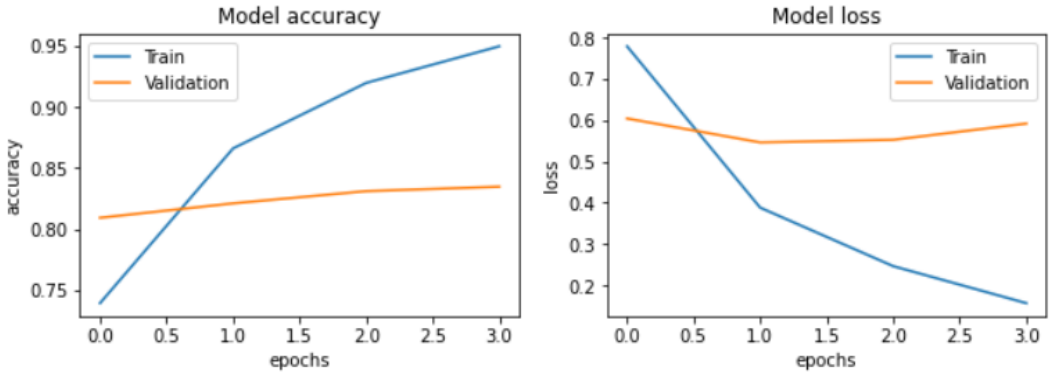

```
Test Loss: 0.69611
Test Accuracy: 76.82%
```

Figure 4**:** EfficientNetB-7(precision, recall, f1-score, support)

ResNet:

The values precision, recall, f1-score, and support. The accuracy of ResNet is 83.07 % and loss is 0.60345.

	precision	recall	f1-score	support
dew	0.86	0.90	0.88	146
fogsmog	0.90	0.90	0.90	173
frost	0.66	0.69	0.67	93
glaze	0.77	0.73	0.75	147
hail	0.79	0.93	0.86	102
lightning	0.93	0.97	0.95	73
rain	0.99	0.64	0.78	107
rainbow	0.97	0.89	0.93	44
rime	0.85	0.84	0.84	245
sandstorm	0.87	0.92	0.90	126
snow	0.68	0.75	0.71	120
accuracy			0.83	1376
macro avg	0.84	0.83	0.83	1376
weighted avg	0.84	0.83	0.83	1376



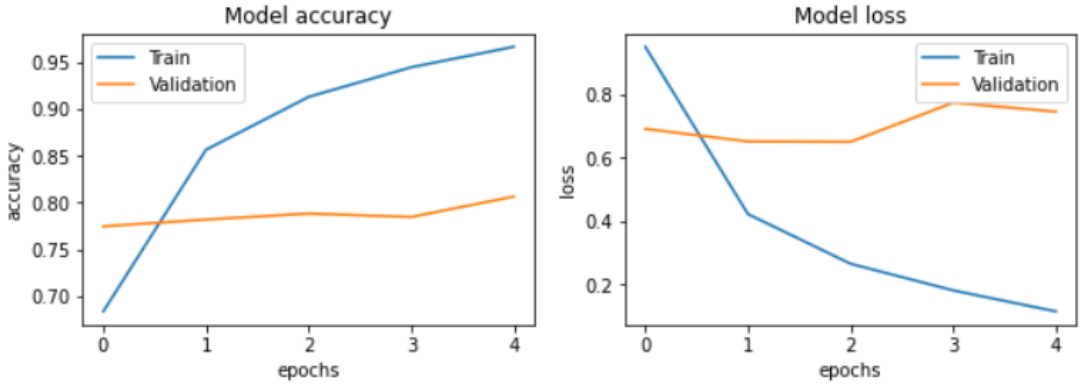

```
    Test Loss: 0.60345
Test Accuracy: 83.07%
```

Figure 5**:** ResNet(precision, recall, f1-score, support)

MobileNet:

The values of precision, recall, f1-score, and support. The accuracy of MobileNet is 83.07 % and loss is 0.61260.

	precision	recall	f1-score	support
dew	0.89	0.90	0.89	146
fogsmog	0.91	0.83	0.87	173
frost	0.69	0.59	0.64	93
glaze	0.71	0.76	0.74	147
hail	0.85	0.92	0.88	102
lightning	0.96	0.93	0.94	73
rain	0.89	0.73	0.80	107
rainbow	0.96	1.00	0.98	44
rime	0.87	0.86	0.86	245
sandstorm	0.81	0.90	0.86	126
snow	0.69	0.77	0.73	120
accuracy			0.83	1376
macro avg	0.84	0.84	0.84	1376
weighted avg	0.83	0.83	0.83	1376



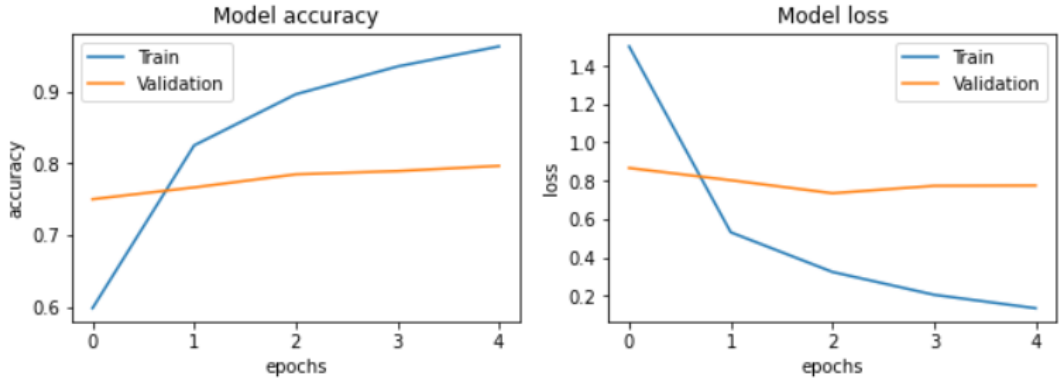

```
    Test Loss: 0.61260
Test Accuracy: 83.07%
```

Figure 6: MobileNet(precision, recall, f1-score, support)

VGG19:

The values of precision, recall, f1-score, and support. The accuracy of VGG19 is 78.92% and loss is 0.83680.

	precision	recall	f1-score	support
dew	0.86	0.82	0.84	146
fogsmog	0.84	0.83	0.83	173
frost	0.54	0.60	0.57	93
glaze	0.68	0.63	0.65	147
hail	0.80	0.87	0.84	102
lightning	0.87	0.95	0.91	73
rain	0.80	0.73	0.76	107
rainbow	0.91	0.93	0.92	44
rime	0.85	0.82	0.84	245
sandstorm	0.85	0.88	0.86	126
snow	0.68	0.71	0.69	120
accuracy			0.79	1376
macro avg	0.79	0.80	0.79	1376
weighted avg	0.79	0.79	0.79	1376



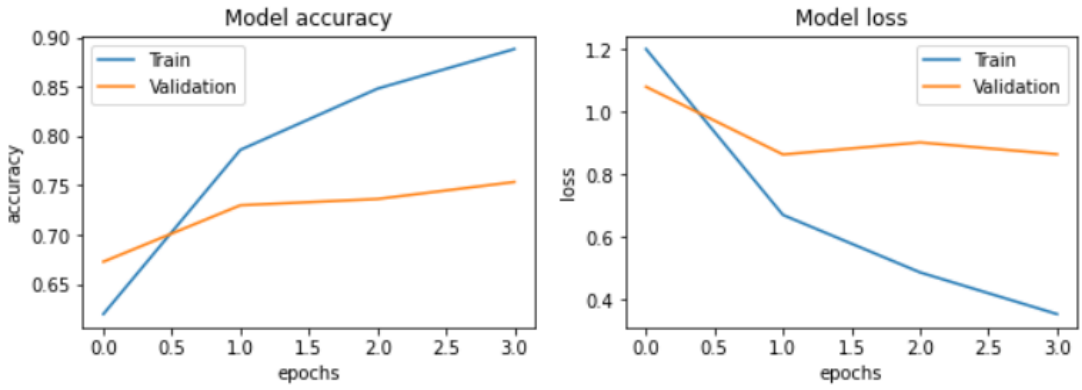

```
    Test Loss: 0.83680
Test Accuracy: 78.92%
```

Figure 7: VGG19(precision, recall, f1-score, support)

Xception:

The values of precision, recall, f1-score, and support. The accuracy of Xception is 76.16% and loss is 0.81368.

	precision	recall	f1-score	support
dew	0.91	0.83	0.87	146
fogsmog	0.80	0.76	0.78	173
frost	0.58	0.65	0.61	93
glaze	0.65	0.64	0.64	147
hail	0.75	0.89	0.81	102
lightning	0.93	0.92	0.92	73
rain	0.81	0.67	0.73	107
rainbow	0.83	0.86	0.84	44
rime	0.77	0.82	0.79	245
sandstorm	0.77	0.83	0.80	126
snow	0.64	0.57	0.61	120
accuracy			0.76	1376
macro avg	0.77	0.77	0.77	1376
weighted avg	0.76	0.76	0.76	1376



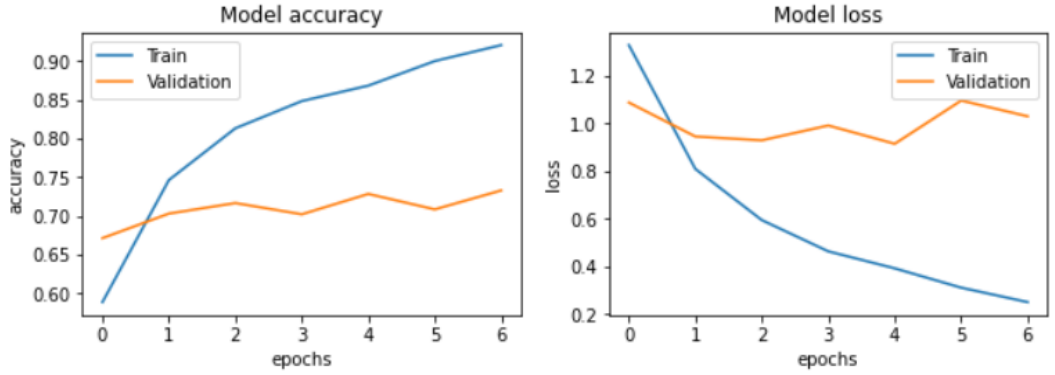

```
Test Loss: 0.81368
Test Accuracy: 76.16%
```

Figure 8: Xception(precision, recall, f1-score, support)

InceptionResNet-2:

The values of precision, recall, f1-score, and support. The accuracy of InceptionResNet-2 is 73.18 % and loss is 0.81368.

	precision	recall	f1-score	support
dew	0.85	0.73	0.79	146
fogsmog	0.79	0.82	0.80	173
frost	0.53	0.63	0.58	93
glaze	0.66	0.54	0.60	147
hail	0.75	0.80	0.77	102
lightning	0.89	0.89	0.89	73
rain	0.72	0.67	0.70	107
rainbow	0.79	0.86	0.83	44
rime	0.71	0.85	0.77	245
sandstorm	0.72	0.76	0.74	126
snow	0.73	0.49	0.59	120
accuracy			0.73	1376
macro avg	0.74	0.73	0.73	1376
weighted avg	0.74	0.73	0.73	1376



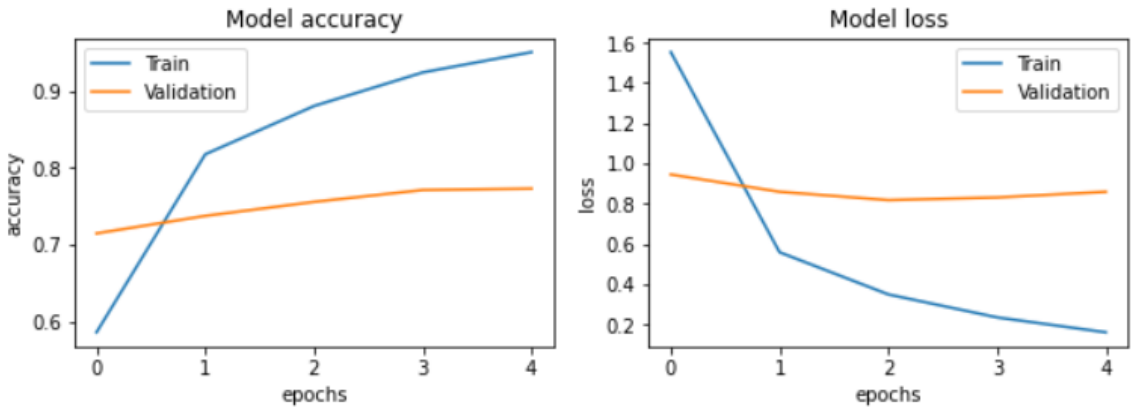

```
Test Loss: 1.12444
Test Accuracy: 73.18%
```

Figure 9: InceptionResNet-2(precision, recall, f1-score, support)

VGG16:

The vales of precision, recall, f1-score, and support. The accuracy of VGG16 is 79.14% and loss is 0.82328.

	precision	recall	f1-score	support
dew	0.95	0.78	0.86	146
fogsmog	0.84	0.90	0.87	173
frost	0.62	0.57	0.59	93
glaze	0.63	0.73	0.68	147
hail	0.76	0.88	0.82	102
lightning	0.93	0.93	0.93	73
rain	0.87	0.69	0.77	107
rainbow	0.91	0.95	0.93	44
rime	0.80	0.82	0.81	245
sandstorm	0.85	0.83	0.84	126
snow	0.68	0.66	0.67	120
accuracy			0.79	1376
macro avg	0.80	0.80	0.80	1376
weighted avg	0.80	0.79	0.79	1376



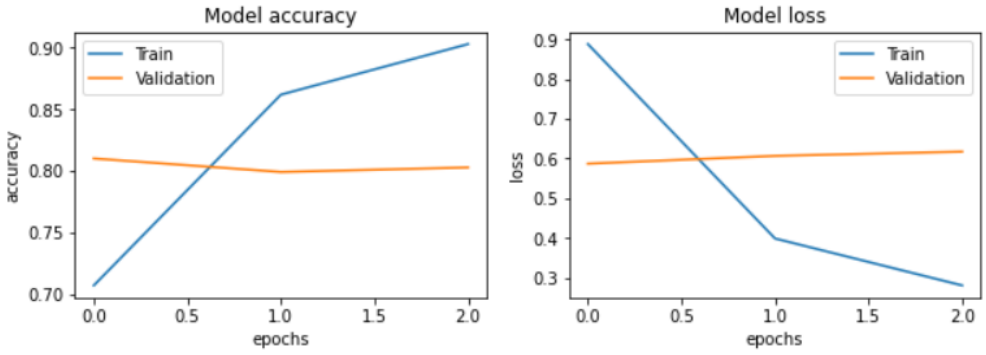

```
    Test Loss: 0.82328
Test Accuracy: 79.14%
```

Figure 10: VGG16(precision, recall, f1-score, support)

ResNet-101:

The vales of precision, recall, f1-score, and support. The accuracy of ResNet-101 is 81.32% and loss is 0.64591.

	precision	recall	f1-score	support
dew	0.95	0.89	0.92	146
fogsmog	0.84	0.96	0.89	173
frost	0.50	0.80	0.61	93
glaze	0.66	0.65	0.66	147
hail	0.96	0.85	0.90	102
lightning	0.87	0.95	0.91	73
rain	0.79	0.79	0.79	107
rainbow	0.88	0.95	0.91	44
rime	0.83	0.88	0.85	245
sandstorm	0.97	0.80	0.88	126
snow	0.92	0.46	0.61	120
accuracy			0.81	1376
macro avg	0.83	0.82	0.81	1376
weighted avg	0.83	0.81	0.81	1376



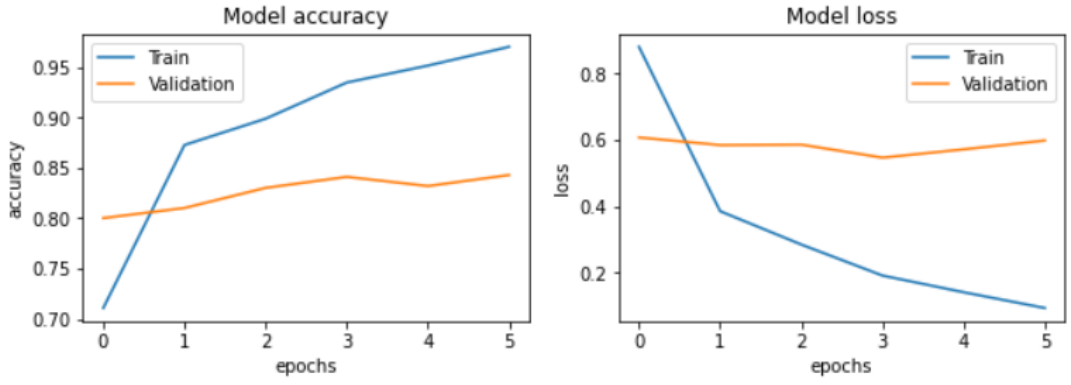

```
Test Loss: 0.64591
Test Accuracy: 81.32%
```

Figure 11: ResNet-101(precision, recall, f1-score, support)

DenseNet-201:

The vales of precision, recall, f1-score, and support. The accuracy of DenseNet-201 is 84.01% and loss is 0.62897.

	precision	recall	f1-score	support
dew	0.94	0.90	0.92	146
fogsmog	0.84	0.95	0.89	173
frost	0.84	0.62	0.72	93
glaze	0.70	0.78	0.74	147
hail	0.84	0.94	0.89	102
lightning	0.93	0.93	0.93	73
rain	0.82	0.79	0.80	107
rainbow	0.89	0.95	0.92	44
rime	0.82	0.87	0.84	245
sandstorm	0.93	0.87	0.90	126
snow	0.79	0.63	0.70	120
accuracy			0.84	1376
macro avg	0.85	0.84	0.84	1376
weighted avg	0.84	0.84	0.84	1376




```
Test Loss: 0.62897
Test Accuracy: 84.01%
```

Figure 12: DenseNet-201(precision, recall, f1-score, support)

Chapter 6

CONCLUSION

The various deep learning methods like EfficientNetB7, ResNet, Mobile Net, VGG19, Xception, InceptionResNetV2, VGG16, ResNet101, DenseNet201 has been applied to this weather recognition project. And it found that DenseNet-201 gave the highest accuracy out of all with 84%.