Weather Classification using Transfer Learning

by
Amalaseeli Arulthasan
Level 4M
Registration No: 2015/SP/011

Supervisor: Dr.M.Siyamalan

A project report submitted in partial fulfillment for the degree of Bachelor of Science
(Specialization in Computer Science)

Faculty of Science
Department of Computer Science
August 2020

University of Jaffna

Abstract

Faculty of Science
Department of Computer Science

Bachelor of Science (Specialization in Computer Science)

By Amalaseeli Arulthasan

This paper presents an empirical analysis of the performance of popular convolutional neural networks (CNNs) for classifying weather category(Sunny, Cloudy, Rainy, Foggy, Snowy) from a given image. The most popular convolution neural networks for object detection and object category classification from images are Alex Nets, GoogLeNet, and ResNet50. A variety of image data sets are available to test the performance of this networks. This study focuses on analyzing the performance of ResNet50. I have taken the most popular kaggle dataset for my study. Moreover I have separated train, validation, and test images for my purposes.

Keywords: Deep Learning, Convolutional Neural Network, Transfer Learning, Neural Network, Weather classification

Declaration of authorship

I, Amalaseeli Arulthasan declare that the report entitled “Weather Classification using Transfer Learning” using Transfer Learning and the work presented in it are done by me under the supervision of Dr.M.Siyamalan, Department of Computer Science, University of Jaffna. I confirm that:

- this work was done wholly or mainly while candidate for a bachelor degree at this University;
- where any part of this report has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated;
- where I have quoted from the work of others, the source is always given;
- I have acknowledged all main sources of help.

Signature of the Candidate **Date**

Certified by:

Supervisor **Date**

Acknowledgement

First and foremost I want to express my gratitude and thanks to my project supervisors, Dr.M.Siyamalan(Department of Computer Science) for his excellent guidance, continuous support and knowledge. He is always ready to help me and suggest ideas based on my movement. This project would not have been completed without his commitment and diligent efforts influenced the content of the project.

Further I would like to thank all academic and non-academic staff members of Department of Computer Science for their immense support and cooperation throughout the study. Moreover, I would like to thank all my friends for their utmost support and motivation.

I cannot end without thanking my friends and parents who have supported and encouraged me through their kindness and affection.

Contents

List of Figures

List of Tables

1 Introduction

In our daily basis of life style affected by weather changes in many ways. Such as solar system, outdoors sporting, agriculture, driver assistance system etc. In recent years, Deep learning is one of the most popular field which is widely used in computer vision and image processing. In modern life weather prediction is most important factor for human beings, this process is very tricky and tough. In some cases, defining sunny and cloudy weather is difficult. Moreover most of the time human beings are able to recognize rainy weather when they are outside. While planning schedule, it is most important to identify weather conditions. Because few day to day activities depend on weather conditions such as choosing dress, transport, etc. Hence, even the human eyes/brain which is advanced and evolved organs can be insufficient to detect or recognize whether conditions from the scene. Even if the problem serves a very wide area to be researched, there is not a sufficient number of or broad projects which are published or in use. Most existing weather classification methods only consider two-class weather conditions such as sunny-rainy or sunny-cloudy weather. All these reasons make me encouraged, curious, and excited about this topic. So I have planned to do the research, on weather classification which is identify the given image sunny, rainy, snowy cloudy or Foggy. To achieve this task I used publicly available kaggle data set. To identify the image weather condition, this data sets have all five category images such as sunny,rainy,cloudy,foggy and snowy.

In recent years, machine learning methods have increased rapidly in various research areas such as robotics, self-automated car system, medical field, etc. Deep Learning(DL), Convolution Neural Networks (CNNs), which allow automatic learning of features from raw images. DL are multi-layered feed-forward neural networks. They consist of simple processing units called artificial neurons, which are organized in layers. Neurons in each layer are connected, through weighted connections, to neurons in the next layers and this cascade of multiple layers is used to automatically extract features that become more complex as the number of layers increases, learning multiple levels of representations that correspond to multiple levels of abstractions. The weights represent the parameters used by the model to transform the input into the output. The output of each intermediate layer can be considered as a representation of the original input data, is it a recombination of the inputs, produced by the previous layers, through these parameters, which are learned to make this recombination useful to finally predict the right output from the input. Hence, to extract always more complex features a large number of layers is required, which involves a corresponding increase in the number of parameters that need to be learned.

What makes CNN different from DL is that it takes into account the spatial structure of an image; indeed, looking at an image, neighboring pixels in a region of the image are more likely to be related than pixels far away CNN's presents also drawbacks, one of this being the high number of meta-parameters that has to be optimally adjusted during the training phase to obtain better performance.

I have used Convolutional Neural Network(CNN) to identify images. In my research, ResNet50 used as pre-trained models. As a computer language TensorFlow used under python with the Keras deep learning library. Also my model is trained validate and tested in cloud environment, Google colab is used as the cloud environment. In my overall project all the images distributed for training 85%, validation 10%, testing 5%.

1.1 Aim/Objective

The aim of this project is to identify weather from the images.

- Exploring the existing work on weather detection as well as Learning state-of-art towards framing a solution for CNN-based weather detection in images.
- Collecting data-sets of labeled images which represent different weather conditions.
- The build CNN architecture used with the Transfer Learning to identify weather conditions of the given images.
- Parameter Tuning is used for increase the performance.

1.2 Contributions

- The Kaggle weather classification data set is used in this project to identify weather conditions. The reason using kaggle data set is containing different weather conditions images such as Sunny, Cloudy, Foggy, Rainy and Snowy.
- ResNet50 pre-trained model is used to achieve the task.

1.3 Report organization

My report is organized as listed bellow:

Chapter 2: Explains the review of relevant literature of the study and this review focuses on method that have been take on by previous researchers and limitations of their methods, as well as a discussion of the results from previous studies.

Chapter 3: Describes the background information that has been used in this research work.

Chapter 4: Explains the proposed methodology that has been used in this project.

Chapter 5: Represent the experimental setup

Chapter 6 Perform testing results and discussions of the proposed method.

Chapter 7: Finally,the conclusions part and explained the work to be do in the future.

2 Related Work

In recent years many researches have done researches based on this weather classification using different methodologies. Nayar [13] developed efficient algorithms for handling rain in computer vision.They first develop a photometric model to describe the intensities produced by individual rain streaks and a dynamic model to capture the spatio-temporal properties of rain.Using both models they developed simple and effective algorithm for detection and removal of rain from videos.

[14], [15], [16] analyzed images taken under poor static weather conditions. They used the atmosphere scattering models, Koschmieder model to estimate scattering coefficients of the atmosphere and restore the contrast of the scene from two or more images taken in bad weather.

Zheng et al [6] classified multi-class weather (sunny,rainy,snowy,haze) which classified images based on five features (Sky, Dark channel, Rain steak, snowflake and Shadow). They used dictionary learning and multiple kernal learning. They used 20K images from MWI(Multi-class Weather image) set. Krizhevsky et al [17] trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. They implemented the special architecture of CNNs for image classification, power needed for its training and overall performance. Their implementation showed the capabilities of an image classifier with a built-in feature extraction based on supervised learning .

In the field of weather classification, CNNs have been used in [7]. In [2], Lu et al. proposed five weather features, i.e., sky, shadow, reflection, contrast, and haze, and proposed a collaborative learning method to classify images into sunny or cloudy. Wei et al [1],Berk et al[10] have classified images for five category of weather conditions(sunny,cloudy,snowy,rainy,and foggy).To enhance driver assistance systems on vehicles, Roser and Mossmann [3] constructed an SVM classifier based on contrast, intensity, sharpness, and color features to classify images captured by the camera mounted on vehicles into clear, light rain, and heavy rain weather conditions. Jacobs and his colleagues[9] explore the relationships between image appearance, sun position, and proposed weather conditions With the Archive of Many Outdoor Scene(AMOS+C) dataset. Similarly, Zhu et al. [8] explored an implementation for the task of image classification, which is the architecture known as GoogLeNet.

Neha sharma[11] have analyzed of Convolutional Neural Network(CNN) for image classification.

Wang et al [23] proposes using recurrent neural networks to capture semantic label dependency and improve multi-label classification. Barsi et al [22] utilized neural networks to examine satellite data in 2003, and the introduction of CNNs sparked a renaissance of training to perform remote sensing tasks. Hung et al [24] used CNNs to identify weeds from a UAV camera, while Hu et al [25] used CNNs to perform remote sensing scene classification, and Chen et al [26] used CNNs to identify vehicles in satellite images.

3 Background

3.1 Introduction Deep Learning

Deep learning is a subset of machine learning.Deep learning, also known as hierarchical learning or deep structured learning.It uses multiple layers to extract higher level features from the input.In this learning models data is filtered through a cascade of multiple layers, with each successive layer using the output from the previous layer.Deep learning models can become more and more accurate as they process more data.

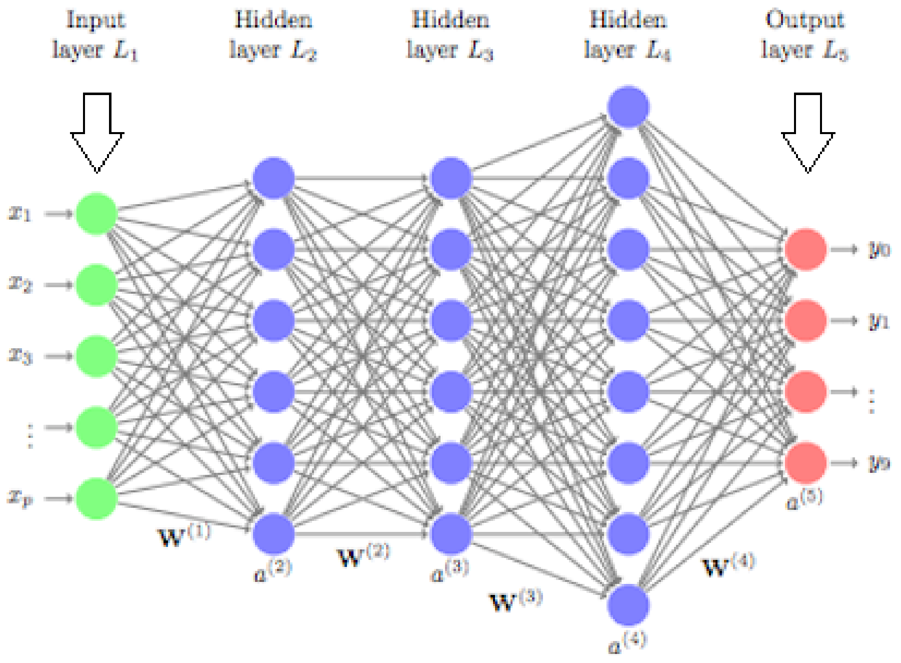


Figure 3.1: Deep Learn architecture

Deep learning requires large amount of labeled data to classify.Deep learning requires substantial computing power. High-performance GPU s have a parallel architecture that is efficient for deep learning.The term “deep” usually refers to the number of hidden layers in the neural network. Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as 150.One of the most popular types of deep neural networks is known as Convolutional Neural Networks(CNN or ConvNet).Number of layers contribute to a model of the data is called *depth* of the model.In deep-learning networks, each layer of nodes trains on a distinct set of features based on the previous layer's output.

3.2 Neural Network

Neural networks are set of algorithm that are designed to recognize patterns.Neural network use for cluster and classify.It works as human brain.Each connection between neurons is associated with a weight. This weight dictates the importance of the input value. The initial weights are set randomly.Each neuron has an Activation Function.Activation functions determine the output of a deep learning model, its accuracy, and also the computational efficiency of training a model.In a neural network, numeric data points, called inputs, are fed into the neurons in the input layer. Each neuron has a weight, and multiplying the input number with the weight gives the output of the neuron, which is transferred to the next layer.There are several kind of neural network such as Multi-Layer Perceptron,Radial basic network,RNN(Recurrent Neural Network),GAN(Generative Adversarial Network),CNN(Convolutional Neural Network)

Neural network consist input,output and hidden layers

- Inout layer:-Receives input data.In our case the input is weather images.Usually input image should be 3D(row,column,color) or more dimensions.
- Hidden layer:-perform mathematical computations on our inputs.One of the challenges in creating neural networks is deciding the number of hidden layers, as well as the number of neurons for each layer.
- Output layer:-returns the output data

3.2.1 Architecture of Neural Network

The basic unit in Neural Network is neuron or node.These units receive input from the external source or some other nodes.It should compute an output based associated weight.Weights to the neuron are assigned based on its relative importance compared with other inputs.

Neural network learn from the input we given.then from the input the layer extract the features and feed it into the next layer.Each layer work as a input layer to another. In Neural network, some inputs are provided to an artificial neuron, and with each input a weight is associated. Weight increases the steepness of activation function. This means weight decide how fast the activation function will trigger whereas bias is used to delay the triggering of the activation function.if the inputs are x1, x2, and x3, then the synaptic weights to be applied to them are denoted as w1, w2, and w3.output is

$$y = f(x) = \sum x_i w_i$$

On the other hand Bias is like the intercept added in a linear equation. It is an additional parameter in the Neural Network which is used to adjust the output along

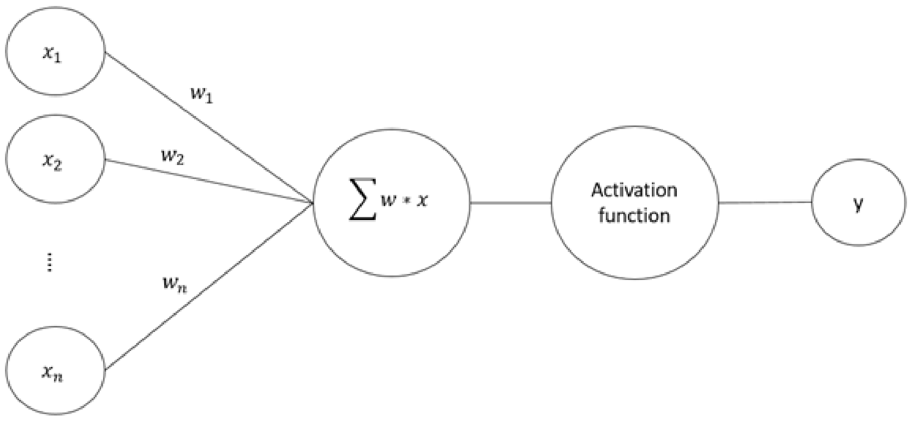


Figure 3.2: Neural Networks

with the weighted sum of the inputs to the neuron. Therefore Bias is a constant which helps the model in a way that it can fit best for the given data.

The processing done by a neuron is thus denoted as

$$y = f(x) = \sum x_i w_i + b$$

where b=bias

3.3 Convolutional Neural Network(CNN)

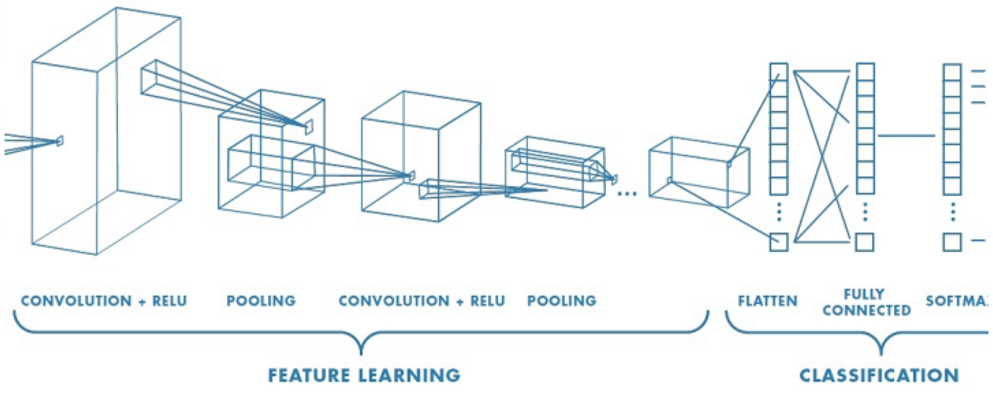


Figure 3.3: General CNN architecture

The convolutional neural network (CNN) is a class of deep learning neural networks. CNN represent a huge breakthrough in image recognition. They're most commonly used to analyze visual imagery and are frequently working behind the scenes in image classification. CNN does is detecting the wanted features from the image data using corresponding filters and extracting the significant features for prediction.

In Figure 3.4 a general ANN architecture is shown. It consists of one input layer, through which the input data x (e.g. the pixel values of an image) is entered into the model, one or more hidden layers and one output layer. The input is propagated forward, layer by layer, until the output layer. A common method for training a neural network is the backpropagation algorithm, used along with an optimization method. The backpropagation algorithm is used to compute the gradient of the loss function, from the output layer and going backwards to the first hidden layer, computing the gradients of each layer's output. These gradients are used by the optimization method to update the parameters in order to minimize the loss function.When there is more than one hidden layer within the network, the architecture takes the name of Deep Neural Network (DNN). Among several Deep Learning architectures, Convolutional Neural Networks are one of the most popular and most promising tools with respect to image classification and analysis.

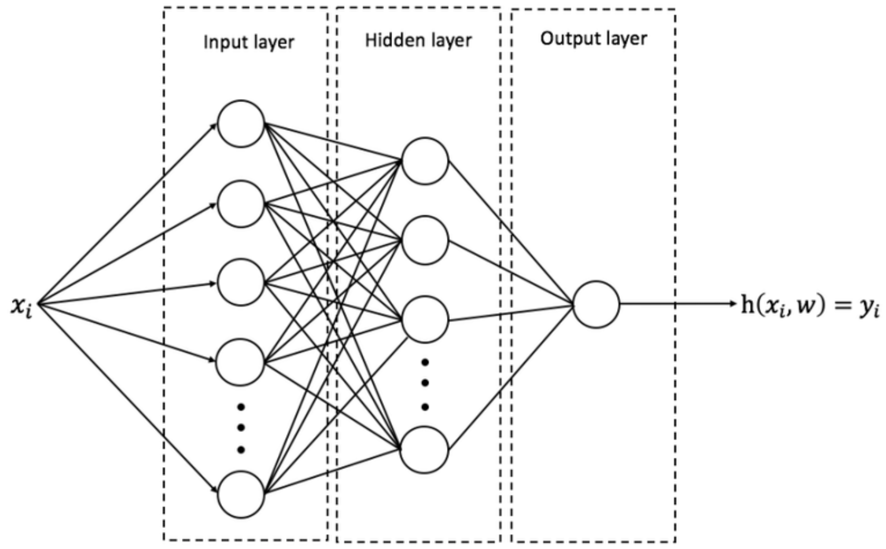


Figure 3.4: General ANN architecture

A classic CNN architecture would look something like this:
Input → Convolution → ReLU → Convolution → ReLU→ Pooling → ReLU → Convolution → ReLU → Pooling → Fully Connected

3.3.1 Architecture of CNN

A CNN architecture consists of layers whose neurons are arranged in 3 dimensions along width, height and depth. Usually, there are three main type of layers that are arranged in sequence to compose the CNN, which are the Convolutional layer, the Pooling layer and the Fully Connected layer

In Figure 3.5 an example of CNN architecture is shown
Convolutional Layer

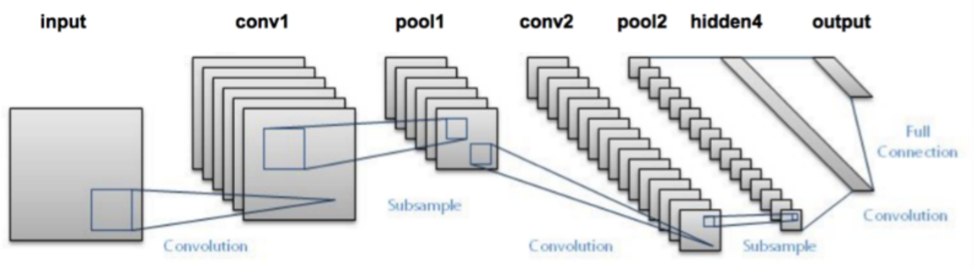


Figure 3.5: Example of CNN architecture

That extracts features from a source image. The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. The parameters of this layer are learnable filters, 3D matrices of numerical values, which, in terms of dimensions, are spatially smaller than the input ones.A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters.

During the forward pass, these filters are slided across width and height of the input at any position. The value with which the filter is slided across the input is called stride.A stride of one means that you're moving your filter over one pixel at a time. The operation of slicing the filters is mathematically translated into a dot product between the filter and the input at any position. The result will be a 2D output that takes the name of activation map, which will be stacked along the depth dimension, together with the other activation maps, to form the output volume. The spatial size of the output can also be controlled by using zero-padding, a technique of adding zeros around the border of the inputs.

There are two types of results to the operation — one in which the convolved feature is reduced in dimensionality as compared to the input, and the other in which the dimensionality is either increased or remains the same. This is done by applying Valid Padding in case of the former, or Same Padding in the case of the latter.The result of this is the convolved feature map. It's smaller than the original input image. This makes it easier and faster to deal with.We create many feature maps to get our first convolutional layer. This allows us to identify many different features that the program can use to learn.

The primary purpose here is to find features in the input image.From the input image, a feature detector, and a feature map. You take the filter and apply it pixel block by pixel block to the input image. You do this through the multiplication of the matrices.then this passes the information on to the next layer.

Pooling layer
A Convolutional layer might be followed by a Pooling layer. Pooling helps to re-

duce the number of required parameters without losing important features or patterns and reduce the amount of computation required. It also helps control overfitting.operating independently on every depth slice. It is a nonparametric layer consisting of filters that slide, with a pre-set value of stride, across the input layer to produce the output. Different can be the functions that the filters are able to perform; the most used are:

- Max Pooling: it takes the maximum value of the input within the filter size.
- Average Pooling: it takes the average value of the input within the filter size.

Fully Connected Layer
Usually, at least one Fully Connected (FC) layer is present in CNN, placed before the network output in order to convert the computed features in class scores.
The fully connected layer is a traditional Multi-Layer Perceptron.Also known as the dense layer.It uses a classifier in the output layer. The classifier is usually a softmax activation function. Fully connected means every neuron in the previous layer connects to every neuron in the next layer. The weights and feature detectors are adjusted to help optimize the performance of the model. Then the process happens again and again and again. This is how our network trains on the data!

3.4 Transfer Learning

From a deep learning perspective, the image classification problem can be solved through transfer learning.Transfer learning is a machine learning technique in which a network that has already been trained to perform a specific task is reused as a starting point for another similar task.when it comes to training a model, Transfer learning is essentially transferring knowledge from one network to another so that you don't have to start from scratch. This can reduce the computational time needed for training drastically.

In computer vision, transfer learning is usually expressed through the use of pre-trained models. A pre-trained model is a model that was trained on a large dataset to solve a problem similar to the one that we want to solve. Accordingly, due to the computational cost of training such models, it is common practice to import and use models from published literature (e.g. VGG, Inception,ResNet MobileNet). A comprehensive review of pre-trained models performance on computer vision problems using data from the ImageNet (Deng et al. 2009) challenge is presented by Canziani et al. (2016). Here in this case the dataset is very small so weights from CNN pretrained on ImagenNet dataset is finetuned to classify weather conditions.

3.4.1 Transfer Learning Process

- Select a pre-trained model: From the wide range of pre-trained models that are available(VGG,ResNet etc). I choose ResNet50 to classify my weather images.

- Classify the problem according to the Size-Similarity Matrix: This matrix classifies the computer vision problem considering the size of dataset and its similarity to the dataset in which the pre-trained model was trained.
- Fine-tune model

3.4.2 ImageNet

It is an open source repository of images which consist of 1000 classes and over 1.5 million images. The dataset which is used for transfer learning is ImageNet. There are many models present in keras such as AlexNet, VGGNet, Inception, ResNet, Xception and many more. In my research I am going to repurpose trained weights from the ResNet50, which is a famous deep neural network, to perform classification on a new dataset.

3.4.3 ResNet

ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8 x 10ˆ9 Floating points operations. It is a widely used ResNet model. Deep Convolutional neural networks are really great at identifying low, mid and high level features from the images.Generally more layers gives better accuracy. When the deeper network starts to converge, a degradation problem has been exposed: with the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly. Such degradation is not caused by overfitting or by adding more layers to a deep network leads to higher training error.

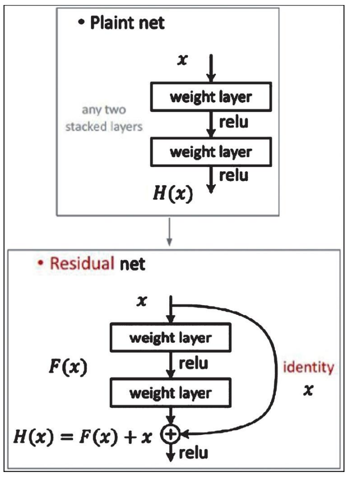


Figure 3.6: Bottleneck Residual Module

To overcome this problem, Microsoft introduced a deep residual learning framework. Instead of hoping every few stacked layers directly fit a desired underlying mapping, they explicitly let these layers fit a residual mapping.The formulation of $F(x) + x$ can be realized by feedforward neural networks with shortcut connections. Shortcut connections are those skipping one or more layers .The shortcut connections perform identity mapping, and their outputs are added to the outputs of the stacked layers

Plain Network

There are two main rules

- For the same output feature map the layers have the same number of filters.

- If the size of the features map is halved, the number of filters is doubled to preserve the time complexity of each layer.

Residual Network

Based on the above plain network, a shortcut connection is inserted. The identity shortcuts $F(xW + x)$ can be directly used when the input and output are of the same dimensions

- The shortcut performs identity mapping, with extra zero entries padded for increasing dimensions. This option introduces no additional parameter.
- The projection shortcut in $F(xW + x)$ is used to match dimensions.

4 Methodology

I used Google's TensorFlow and the open-source neural network Python package Keras for the majority of my model development.From the literature study presented in Chapter 3, observed that CNN based approaches are the stateof-art among other image recognition methods [18]. In particular, ResNet [19] is the most desirable architecture for our research project. The aim of this project is to detect weather from the image. This can be done in various way. First way is to form a large training set for the model.Second, need to train the network on the data collected. For this purpose use Transfer Learning on a predefined architecture. In my research I am using ResNet50 architectures, trained on the imagenet dataset, and fine-tune its weights on the newly generated dataset. Then optimize this model so as to improve its accuracy by limiting overfitting.

My design steps are as follows: First I developed a basic CNN architecture, Then build ResNet50 model[20].

4.1 ResNet50

4.1.1 ResNet archtecture

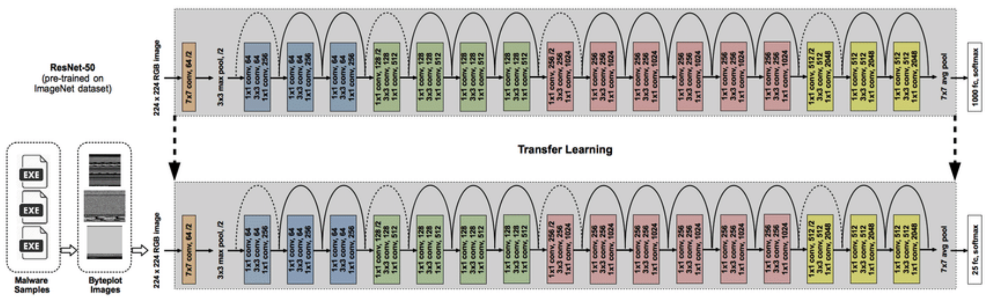


Figure 4.1: ResNet50 architecture

4.1.2 ResNet architecture

As you can see in figure 3.6 explained ResNet50 architecture.

- A convoultion with a kernel size of 7 * 7 and 64 different kernels all with a stride of size 2 giving us 1 layer.
- Next the max pooling layer which also a stride size of 2.
- In the next convolution there is a 1 * 1,64 kernel following this a 3 * 3,64 kernel and at last a 1 * 1,256 kernel, These three layers are repeated in total 3 time so giving us 9 layers in this step.
- Next can see kernel of 1 * 1,128 after that a kernel of 3 * 3,128 and at last a kernel of 1 * 1,512 this step was repeated 4 time so giving us 12 layers in this step
- After that there is a kernal of 1 * 1,256 and two more kernels with 3 * 3,256 and 1 * 1,1024 and this is repeated 6 time giving us a total of 18 layers
- Then again a 1 * 1,512 kernel with two more of 3 * 3,512 and 1 * 1,2048 and this was repeated 3 times giving us a total of 9 layers.
- After that average pool and end it with a fully connected layer containing 1000 nodes and at the end a softmax function so this gives us 1 layer. We don't actually count the activation functions and the max/ average pooling layers.

  so totaling this it gives us a 1 + 9 + 12 + 18 + 9 + 1 = 50 layers Deep Convolutional network. This architecture can be used on computer vision tasks such as image classififcation, object localisation, object detection.

4.2 Implementation

At the beginning,I researched and examined lots of related work and paper on this task or similar weather recognition tasks. As stated in my introduction, my aim is to develop a program that can classify weather condition from given image. I researched a long time to find out how to solve this problem more generally with high accuracy.

When training dataset is relatively small, transferring a network pretrained on a large annotation dataset and fine-tuning it for a particular task are an efficient way to achieve acceptable accuracy and less training time. It encourages me to do my research work with the best pre-trained model and then do the parameter tuning[21]. To achieve this task I choosed ResNet50. As a computer language TensorFlow used under python with the Keras deep learning library. I collected datasets and images for 5-class weather from kaggle . Then I split them for testing and organize for general use. Also my model is trained validate and tested in cloud environment with this kaggle dataset., Google colab is used as the cloud environment.

I used categorical cross-entropy loss function and the Adam optimizer. During training, I used Keras callbacks to save model weights when the model's performance on tracked metrics. The entire labeled images could not be loaded all at once, even with 120 GB of memory on our Google Cloud instance because of large amount of data. To remedy this, I used Keras' model.fit generator method, which loads batches of images from a generator and fits the model. The ImageDataGenerator class allows for easy image resizing to fit the current model.

5 Experimental Setup

5.1 Dataset

There are different datasets for detecting different weather conditions. The Image2Weather dataset consists of more than 180,000 images of global landmarks of four weather categories such as sunny, cloudy, rainy, snowy, and foggy[1]. However, the images used for training are still limited and represent cities during the daytime. Accordingly, data augmentation techniques have been applied to enhance the training of each model. The datasets are augmented by rescaling, shearing, horizontal flips, and zooming.

Similarly, the Multi-class Weather Image (MWI) dataset consists of 20,000 images of different weather conditions[6]. Additionally, a large dataset of images is presented to describe weather conditions from the aspect of cloud intensity such as clear, partly cloudy, mostly cloudy, or cloudy including the time and location data[9]. Another example is a binary weather dataset that contains 10,000 images belonging to either sunny or cloudy weather[5].

The dataset was created, images are downloaded from kaggle and saved in the local disk which contains weather depicting images For the purpose of train ResNet50 model. The dataset was labeled by identifying objects in the images such as sunny, cloudy, rainy, foggy and snowy. Then manually I created 85% Testing images,10% for Validation and 5% for Testing images. Training set includes 2484 labeled images, validation set contains only 257 images. Figure 3.8 shows a sample of the multi-class images for training, testing, and validation. Images are not fixed dimensions and the photos are different sizes.

Number of Total images	
Training	Validation
2484	257

Table 5.1: Total number of Datasets

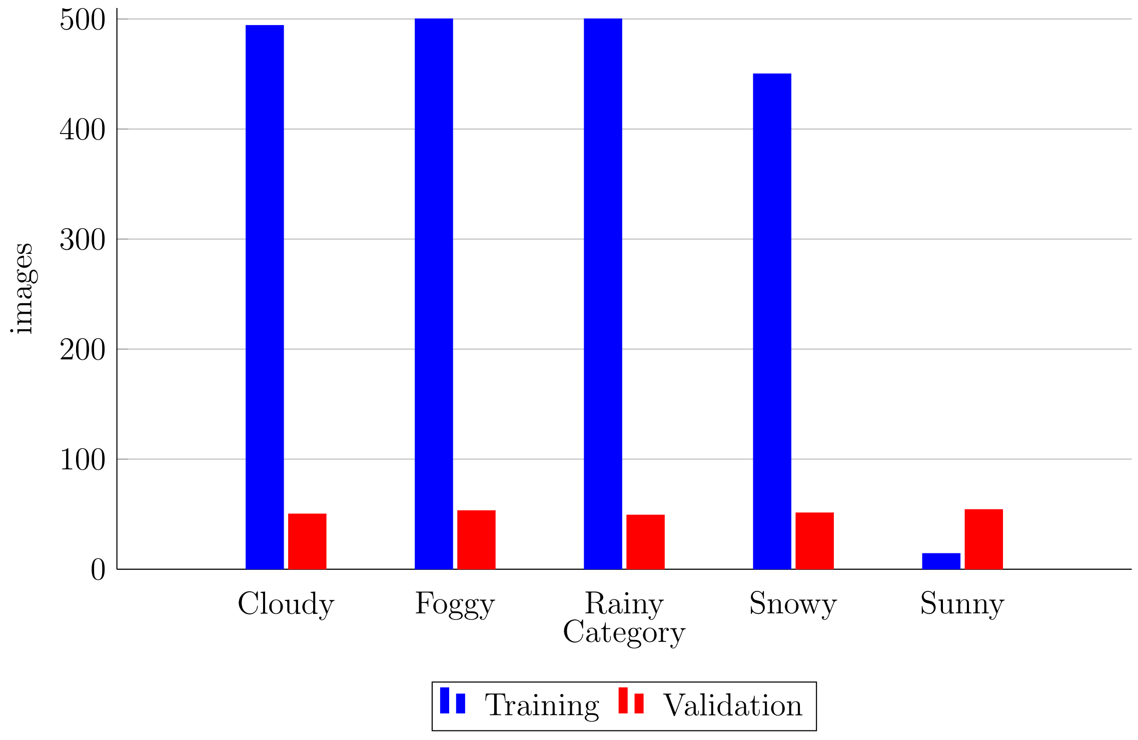
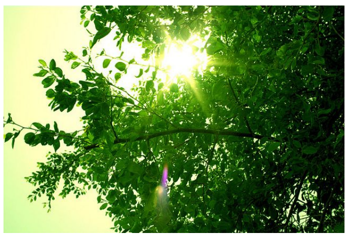


Figure 5.1: Distribution of image classes

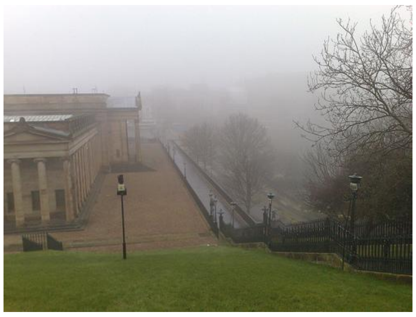
5.2 Parameter Tuning

Fine tuning refers to the process of training the last few or more layers of the pre-trained network on the new dataset to adjust the weight. I used ResNet50 pretrained model with kaggle dataset. I loaded the model with the same pretrained ImageNet weights. I added Global Average Pooling and a dense output layaer to the ResNet-50 model. Initiate GlobalAveragePooling2D which eliminate the spatial dimensions. The aim is to add Global Average Pooling is to have various size of images in my work. I have added 4 dense layer with 1024,1024,1024,512 neurons respectively.

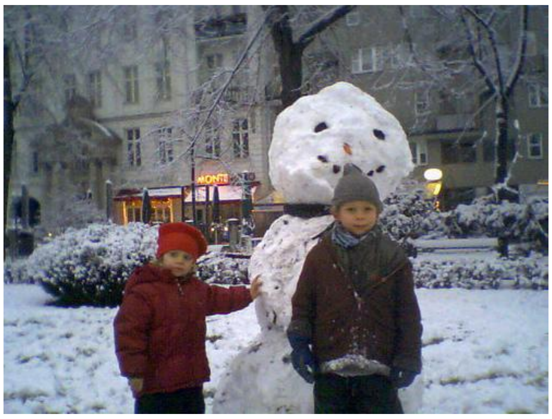


(a) Sunny

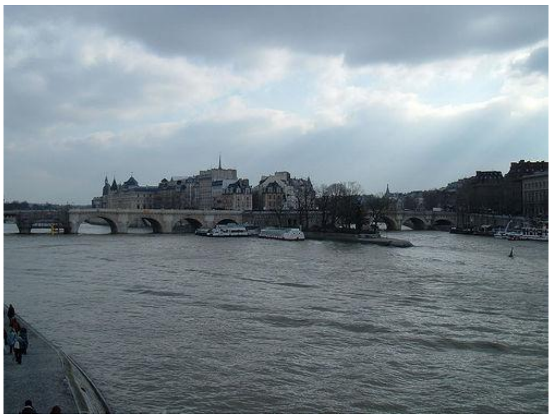
(b) Rainy



(c) Foggy



(d) Snowy



(e) Cloudy

Figure 5.2: The sample images of Dataset

6 Results and Discussion

I started by evaluating the performance on ResNet-50 applying the fine-tuning approach, using as initial weights the one of the network trained on the ImageNet dataset, The Keras ResNet got to an accuracy of 80.0% after training on 5 epochs with Adam optimizer. With the learning rate of 0.001 Keras ResNet got to an accuracy of 77.0%. For training the model I choosed 2484 images from kaggle dataset.

Component of model	
CNN model	ResNet50
Optimizer	adam
loss	categorical-crossentropy
batchsize	50
epochs	5

Table 6.1: Initial Component

Training loss	Training accuracy	Validation loss	validation accuracy
22%	91%	54 %	81%

Table 6.2: Initial model Performance

The confusion matrix of weather type classification based on the ResNet50 model

	Sunny	Rainy	Cloudy	Foggy	Snowy
Sunny	0.80	0.04	0.10	0.03	0.04
Rainy	0.16	0.98	0.33	0.33	0.32
Cloudy	0.045	0.071	0.099	0.039	0.086
Foggy	0.01	0.02	0.01	0.09	0.02
Snowy	0.01	0.02	0.04	0.01	0.91

Table 6.3: Confusion matrix for predicted test dataset.

For learning rate: 0.001 the model got accuracy only 77%

Validation loss	validation accuracy
68%	77%

Table 6.4: model performance with learning rate 0.001

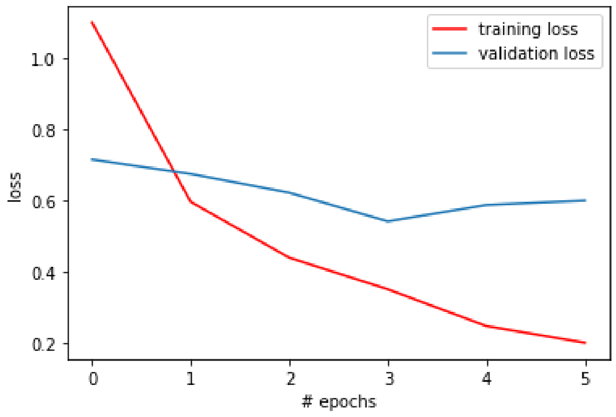


Figure 6.1: Model loss

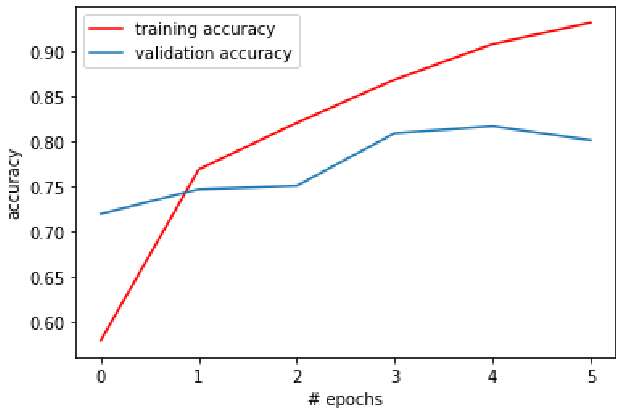
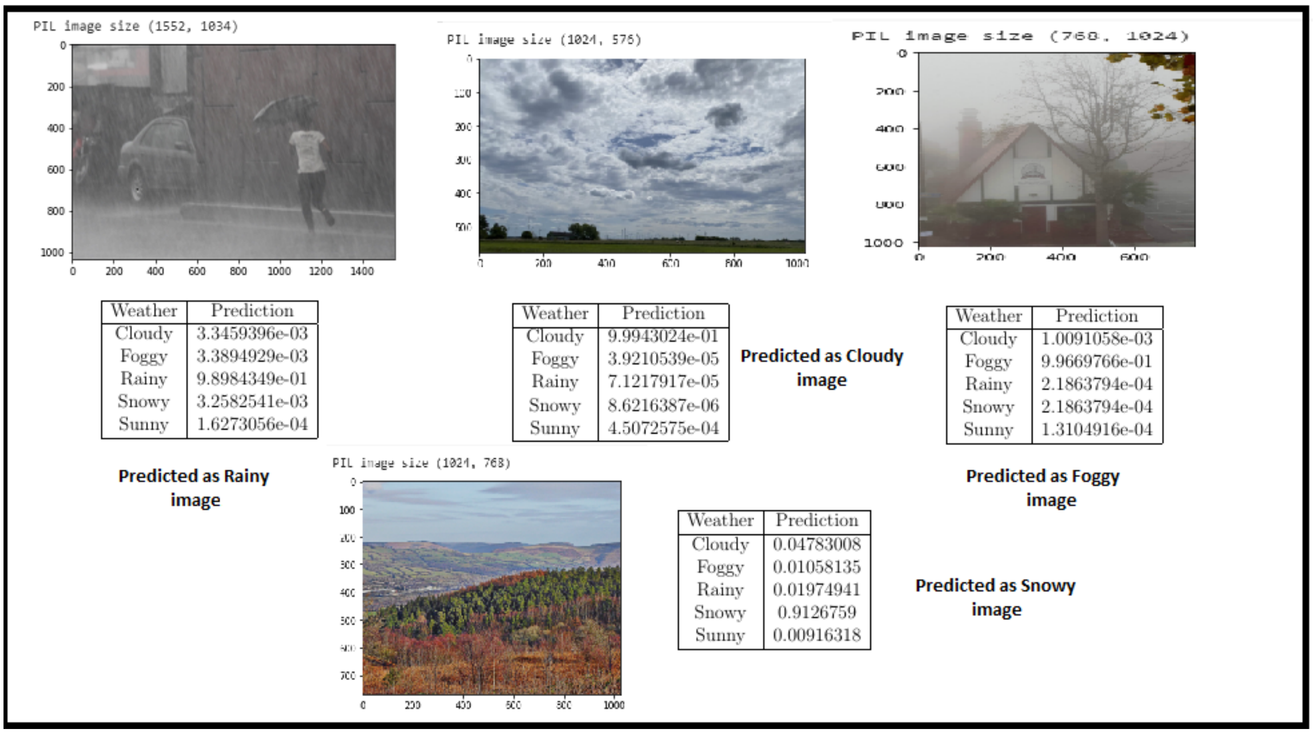


Figure 6.2: Model Accuracy

Figure 6.3: Learning Curve ResNet-50 for baseline model


PIL image size (1552, 1034)
PIL image size (1024, 576)
PIL image size (768, 1024)
PIL image size (1024, 768)

Weather	Prediction
Cloudy	3.3459396e-03
Foggy	3.3894929e-03
Rainy	9.8984349e-01
Snowy	3.2582541e-03
Sunny	1.6273056e-04

Predicted as Rainy image

Weather	Prediction
Cloudy	9.9943024e-01
Foggy	3.9210539e-05
Rainy	7.1217917e-05
Snowy	8.6216387e-06
Sunny	4.5072575e-04

Predicted as Cloudy image

Weather	Prediction
Cloudy	1.0091058e-03
Foggy	9.9669766e-01
Rainy	2.1863794e-04
Snowy	2.1863794e-04
Sunny	1.3104916e-04

Predicted as Foggy image

Weather	Prediction
Cloudy	0.04783008
Foggy	0.01058135
Rainy	0.01974941
Snowy	0.9126759
Sunny	0.00916318

Predicted as Snowy image

Figure 6.4: Predicted results

Due to the strong overfitting, which can be inferred both from the distance between the training and the validation accuracy curves and the increasing trend of the validation loss curve, we decided to act on the regularization, studying the effects that the addition of the full pre-activation framework and dropout would have had on the network. The 17 pre-trained neural networks were trained on a large data set by using more than a million images, as a result the algorithms developed can classify new images into 1,000 different object categories.Through artificial intelligence and machine learning each network can detect images based on unique

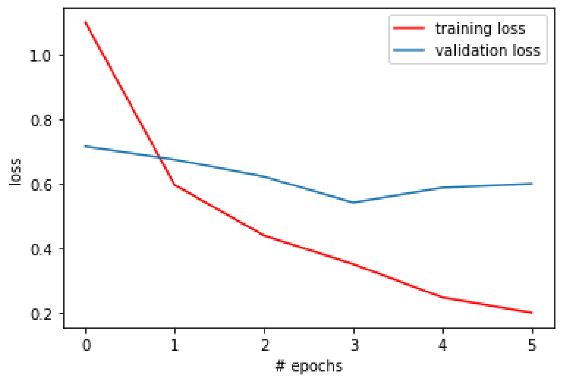


Figure 6.5: Model loss

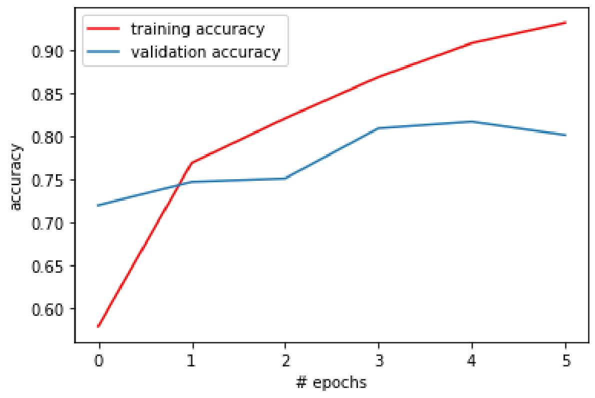


Figure 6.6: Model Accuracy

Figure 6.7: Learning Curve ResNet-50 with Dropout with weight decay 0.001

features representative of a particular category.

Keras Image generator generate training data from the directories/ numpy arrays in batches and processes them with their labels. Traing data was also shuffled during training the model, while validation data was used to get the validation accuracy and validation loss during training.

As seen from confusion matrix, this model is really good at predicting all the classes.

7 Conclusion and Future work

Deep learning models like convolutional neural networks (ConvNet) require large amounts of data to make accurate predictions. In general, sufficient sample size for a ConvNet application would involve tens of thousands of images. Often, only a few thousand labeled images are available for training, validation, and testing. Challenges associated with limited data can be overcome if you leverage pre-trained layers. Keras pretrained models can be used for prediction, feature extraction, and fine-tuning. ResNet is a powerful backbone model that is used very frequently in many computer vision tasks.

In this paper I proposed and classified five weather conditions from images using transfer learning technique. I proposed ResNet50 as my model.Additionally, combining transferred features from multiple CNNs could further improve the classification accuracy. Due to time and computational cost it was not possible for me to run more experiments using different known architectures other than ResNet50.

In fututre work, compare the performance of keras models such as ResNet, Inception, VGG pretrained on imagenet dataset. I'll be using all of them and comparing them for best accuracy. It's definitely possible that a different architecture would be more effective.

8 References....

[1]Wei-Ta Chu,Xiang-You Zheng,Ding-Shiuan Ding,"*Image2 Weather:A Large-Scale Image Dataset forWeather Property Estimation,National Chung Cheng University,*"IEEE 2016.

[2]Cewu Lu,Di Lin,Jiaya Jia,Chi-Keung Tang,"*Two-class Weather Classification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3718–3725

[3]Martin Roser,Frank Moosmann,"*Classification of Weather Situations on Single Color Images," in Intelligent Vehicles Symposium, 2008 IEEE.* IEEE, 2008, pp. 798–803..

[4]Jose Carlos Villarreal Guerra1, Zeba Khanam1, Shoaib Ehsan1, Rustam Stolkin2, Klaus McDonald-Maier1,"*Weather Classification:A new multi-class dataset,data augmentation approach and comprehensive evaluations of Convolutional Neural Network*",arXiv:1808.00588v1 [cs.CV] 1 Aug 2018.

[5]C. Lu, D. Lin, J. Jia, and C. K. Tang, "*Two-class weather classification,*", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2510–2524, Dec 2017.

[6] Zheng Zhang and Huadong Ma,Huiyuan Fu, Cheng Zhang, "*Scene-free multi-class weather classification on single images," in Image Processing (ICIP), 2015 IEEE International Conference on. IEEE,* 2015, pp. 4396–4400

[7] Mohamed Elhoseiny, Sheng Huang, and Ahmed Elgammal, "*Weather classification with deep convolutional neural networks," in Image Processing (ICIP), 2015 IEEE International Conference on.* IEEE, 2015, pp. 3349–3353.

[8]Ziqi Zhu, Li Zhuo, Panling Qu, Kailong Zhou, and Jing Zhang, "*Extreme weather recognition using convolutional neural networks," in Multimedia (ISM), 2016 IEEE*

International Symposium on. IEEE, 2016, pp. 621– 625

[9] M. Islam, N. Jacobs, H. Wu, and R. Souvenir, *"Images+weather: Collection, validation, and refinement," in Proceedings of IEEE CVPR Workshop on Ground Truth, 2013".*

[10] Berk Gulay,Samet Kalkan,Mert Surucuoglu "*Weather Condition Prediction from Image"*

[11]Neha Sharma,Vibhor Jain, Anju Mishra, *"An Analysis Of Convolutional Neural Networks For Image Classification," International Conference on Computational Intelligence and Data Science (ICCIDS 2018)*

[12]Heather Blundell, Rickard Bruel Gabrielsson, Dylan Liu, *" Network Analysis of Convolutional Neural Networks' Weights on Images"*

[13]K. Garg and S.K. Nayar. *"Vision and rain. International Journal of Computer Vision",* 75(1):3–27, 2007.

[14]S.G. Narasimhan and S.K. Nayar. *"Chromatic framework for vision in bad weather. In IEEE Conference on Computer Vision and Pattern Recognition"*, volume 1, pages 598–605, 2000.

[15]S.G. Narasimhan and S.K. Nayar. *"Removing weather effects from monochrome images. In IEEE Conference on Computer Vision and Pattern Recognition"*, volume 2, pages 186–193, 2001

[16] S.G. Narasimhan and S.K. Nayar. *"Vision and the atmosphere. Internatl. Journal of Computer Vision"*, 48(3):233–254, 2002

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, *"Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems"*, 2012, pp. 1097–1105

[18] S. Srinivas, R. K. Sarvadevabhatla, K. R. Mopuri, N. Prabhu, S. S. Kruthiventi, and R. V. Babu, A taxonomy of deep convolutional neural nets for computer vision, arXiv preprint arXiv:1601.06615 (2016).

[19] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, arXiv preprint arXiv:1512.03385 (2015).

[20] Daniel Gardner, David Nichols *"Multi-label Classification of Satellite Images with Deep Learning"*

[21]J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in Proceedings of the 28th Annual Conference on Neural Information Processing Systems 2014, (NIPS '14), pp. 3320–3328, December 2014.

[22] A. Barsi and C. Heipke, "Artificial neural networks for the detection of road junctions in aerial images," International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, vol. 34, no. 3/W8, pp. 113–118, 2003.

[23] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2285–2294.

[24] C. Hung, Z. Xu, and S. Sukkarieh, "Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a uav," Remote Sensing, vol. 6, no. 12, pp. 12 037– 12 054, 2014.

[25] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," Remote Sensing, vol. 7, no. 11, pp. 14 680–14 707, 2015.

[26]X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," IEEE Geoscience and remote sensing letters, vol. 11, no. 10, pp. 1797–1801, 2014.