# WEATHER CLASSIFICATION FROM STILL IMAGES USING ENSEMBLE METHOD

# BY

**AJAYI GBEMINIYI OLUWAFEMI**
**STUDENT NO: 50374443**

**submitted in accordance with the requirements for the degree of**

**MAGISTER TECHNOLOGIAE**

**In the subject**

**Electrical Engineering**

**at the**

**University of South Africa**

**Supervisor: Professor Zenghui Wang**

**January 2021**

# DECLARATION

**Name:** AJAYI Gbeminiyi Oluwafemi

**Student number:** 50374443

**Degree:** Magister Technologiae (MTech): Electrical Engineering

Exact wording of the title of the dissertation or thesis as appearing on the copies submitted for examination:

Weather Classification from Still Images Using Ensemble Method

I declare that the above dissertation/thesis is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

31/01/2021

SIGNATURE DATE

# ACKNOWLEDGMENT

First and the foremost, I would like to express my sincere gratitude to God almighty for the grace to complete this project. Secondly, I would like to thank my research supervisors – Professor Zenghui Wang from the Department of Electrical and Mining Engineering at the University of South Africa (UNISA), for his guidance, patience and the motivation he provided to achieve the goals of this research. I cannot imagine better supervisor for research study than him.

My sincere thanks go to my friends, Idowu Seriki, for his constant encouragement and love he showed during the research work.

Finally, I would like to thank my beloved spouse; Uzoamaka Ajayi and my parents, and the rest of my family for their love and support not just during the research work, but also throughout my life.

# PUBLICATION

1. Gbeminiyi Ajayi and Zenghui Wang, "Multi-class Weather Classification from still Images using Ensemble Method" Proceeding of the 2019 Southern African Universities Power Engineering Conference/ Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA), Bloemfontein, 29 - 31 January 2019, Page 135-140.
2. Mendeley Data - Multi-class Weather Dataset for Image Classification http://dx.doi.org/10.17632/4drtyfjtfy.1#file-b3b8a956-4bcb-431d-bda0-d466af180d2e

# ABSTRACT

In the field of computer vision, multi-class outdoor weather classification is a difficult task to perform due to diversity and lack of distinct weather characteristic or features. This research proposed a novel framework for identifying different weather scenes from still images using heterogeneous ensemble methods. The approach was based on construction of unobstructed opaque cloud coverage (OCC) multi-class weather images; and the introduction of diversity concept called Selection Based on Accuracy Intuition and diversity (SAID) for the construction of stacked ensemble models. The stages involve the extraction of histogram of features from different weather scenes to determine their contribution to the overall performance of the experiment, training and validating the performance of the model. The blending and boosting of different weather features using stacked ensemble algorithms shows an average accuracy of over 90% in recognizing rainy still images and over 80% for sunny, sunrise and sunset still images. Similarly, the meta-learner of the stacked ensemble model performed better than the individual base learners of the model. The research presents academic and practitioners a new insight into diversity of heterogeneous stacked ensemble methods for solving the challenges of weather recognition from still images.

**Key words:** Computer vision; Image classification; Stacking ensemble; ensemble diversity; weather identification; recognition; machine learning; image preprocessing; feature extraction; heterogenous concept

# TABLE OF CONTENT

# LIST OF FIGURES

# LIST OF TABLES

# List of Acronyms

| | |
|---|---|
| **MKL** | Multiple kernel learning |
| **SVM** | Support Vector Machine |
| **KNN** | K-Nearest Neighbours |
| **BMA** | Bayesian Model Averaging |
| **MLR** | Multi-Response Linear Regression |
| **MWD** | Multi-Class Weather Dataset |
| **SAID** | Selection based on Accuracy Intuition and Diversity |
| **OCC** | opaque cloud coverage |
| **HSV** | Hue, Saturation and Value |
| **LBP** | Local Binary Pattern |
| **BVC** | Bias Variance Covariance |
| **API** | Application programming interface |
| **RAM** | Random access memory |

# CHAPTER ONE
# INTRODUCTION

## 1.0 BACKGROUND

Weather is an essential component in human life. It determines how and where we live, what we eat and what we wear. In fact, it controls our time. Understanding different weather conditions is tantamount to taking control of our time and future. In fact, the knowledge of weather is of great importance to farmers, pilots, marine traffic marshals and so on.

Recently, extreme weather conditions have led to natural disasters like floods, landslides, disruption in communication and transportation system, loss of properties, crop, livestock and damage of infrastructure and building (Fischer and Knutti, 2015).

For years weather forecasting has been based on quantitative data collected through different instruments and tools. Generally, Sensors has been the major instrument or device used by meteorologists for weather observation and detection in most weather stations. However, these are to purchase and maintain especially for less developed nations costly (Lu, Lin, Jia, and Tang 2014). However, in this era of Internet of Things (IoT) using a digital camera like surveillance camera or phone camera connected to computer to observe our environment for weather recognition might be cost effective and produce an intelligent computer vision system.

Several computer vision researches have been proposed to address this issue however, majority of the recognised researches on weather are postulated based on clear weather assumption (Nashashibi, Charette, and Lia 2010), such as driver assistance systems (Kurihata, Takahashi, Ide, Mekada, Murase, Tamatsu and Miyahara, 1963), video surveillance (Woo, Jung, Kim, and Seo, 2010) and robot navigation (Katsura, Miura, Hild, and Shirai 2003) which are high affected by the dynamic nature of weather. The bad weather conditions usually affect the quality of images or videos, and it is necessary to use the weather information to correct the image/video processing algorithms to achieve better performance.

Weather classification from still images is usually time consuming, difficult and requires experience skills and knowledge to identify distinct weather features. The extant literature indicates that researchers have in the past used multiple kernel learning (MKL) to determine the optimal weather features combination for weather classification. This method discards features that are weak to contribute to the overall recognition. However, this research work addresses the problem of weather features combination and recognition using the state-of-art ensemble methods.

## 1.2 PROBLEM STATEMENTS

In general, sensors-based devices are used at weather stations by meteorologist to recognize different weather condition which is costly. However, with intelligent computer vision application using simple digital camera can reduces cost significantly.

In a computer vision, multi-class outdoor weather classification also poses a difficult task to perform due to diversity and lack of distinct weather characteristic or features. This research also presents insight into diversity of ensemble heterogenous method for solving problems associated with weather recognition problem.

## 1.3. RESEARCH GOAL

The main goal of this research is as follows:

- Collect a dataset containing images showing different weather condition with their corresponding labels indicating the specific weather condition.

- to propose a novel framework for recognizing different weather condition through the eyes of digital cameras that can be used at weather stations for weather forecast.
- Compare the proposed method with other well-known classification methods

## 1.4. RESEARCH QUESTIONS

Research questions are formulated to make the research problem tractable with limited scope, and to ensure that the resulting model is as useful as possible in real-world applications. The main research question is:

*How is an intelligent computer vision system developed that can accurately recognize different weather condition that can replace sensor devices?*

To answer the above research question, the following sub-questions were formulated to answer the questions.

1. What multi-class weather dataset is available?
2. How to select distinct weather features?
3. What weather diversity method would be employed for better performance?
4. What model would be employed for better performance

These research questions will guide the background study, review of related literatures, research methodology and dataset collection methods.

## 1.5. RESEARCH ASSUMPTION

Since there is no standard way of measuring diversity of ensemble models, the research will adopt 10% accuracy difference between different classification models for selection models that will make up the ensemble method.

## 1.6. RESEARCH LIMITATION

This research work encountered similar challenges that are common to most computer vision research works such as

a. Dataset Availability: Because there are few or no obstructive multi-class weather dataset, the dataset was manually collected online and annotated; and

b. Computation Resource: The experiment will be performed on a laptop computer with configuration of intel Core i5, clock-speed of 1.2GHz and 8GB RAM. Therefore, high dimensional dataset images will be preprocessed to fixed dimension of 128 for each feature.

## 1.7 RESEARCH METHODOLOGY

The research methodology for this dissertation involves the following steps to be taken:

a. Experiment/Coding environment setup: This stage involves identification and configuration of tools required to perform detail analysis that will answer the research questions in Section 1.4.

b. Dataset: This step involves the collection and preprocessing of images of different weather conditions.

c. Feature Extraction: This involves the extraction distinct weather features for different weather conditions.

d. Model Development: This stage focuses on the design and programming development of stacked ensemble algorithms that will be used for training and testing of the model performance.

e. Analysis and Interpretation of result: This is last stage of the research which involves analysis and interpretation of model performance

## 1.8 RESEARCH CONTRIBUTION

Since the emergence of image classification, the challenge of selecting distinct weather features for effective recognition is still an open challenge that requires further attention. Hence, the contribution of this dissertation is aim at addressing this challenge, by proposing and empirically verifying alternatives that may be used to find distinct weather features for weather classification. The specific contribution and departure points from previous works are as follows:

a. Alternative method of selecting distinct weather features;

b. Weather Dataset: In the absent of public unobtrusive weather dataset, the dissertation work was able gather suitable weather images for classification task;

c. Introduction of new concept to combination of base learners that is based on Selection Based on Accuracy and Intuition Diversity (SAID); and

d. The first application concept for stacked ensemble method for classification of weather from still images.

## 1.9. DISSERTATION OUTLINE

This dissertation consists of five chapters, which are closely linked to the research objectives discussed in Section 1.3. They are structured as follows:

- **Chapter 1**: This presents the general overview of the dissertation by highlighting related research work and its shortcomings, the objectives or goals of the project and methodology that guides the implementation of this project.

- **Chapter 2**: This chapter provides the background reviews or survey on the weather classification techniques and the reasons for image classification task. It also provides focus for the dissertation research by highlighting the current challenges faced by researchers in the task of weather classification techniques. This followed by background to the proposed ensemble stacked techniques.

- **Chapter 3**: In this chapter, a background method of applying stacked ensemble method to the task of selecting distinct weather features, base learner diversity and meta-learner is formalized, a step-by-step method of setting-up experimental tools and configuration of the research proposed method.

- **Chapter 4**: This chapter takes focuses on carrying out different experiment from pre-processing of different weather images to feature extraction and selection, the measurement for diversity of the base-learners and its influence on the overall experiment. Thereafter, the experiment results are discussed in detail.

- **Chapter 5**: This is the last chapter of the dissertation where the research study is summarized, recommendation and concluded. The research limitation and future work were also highlighted.

# CHAPTER TWO

# BACKGROUND AND RELATED STUDY

## 2.0 CHAPTER OVERVIEW

This chapter examine the past research works or studies relating to the subject matter of this dissertation work. The first part of the chapter focusses on the extant review of related literatures or past research works on weather recognition as presented in Section 2.1. In section 2.2 and 2.3, a discussion on ensemble method, the combination of algorithms, and its applications are given in an in-depth manner. Section 2.4 gives an outline of the specific ensemble method used for phenomenon under study. Lastly, a review of image processing techniques used in this research is analyzed.

## 2.1 WEATHER RECOGNITION FROM STILL IMAGES

Several researchers have attempted to classify different weather conditions from images or videos using different machine learning and image processing techniques. To review them, we start by examining the basic discriminative feature of different weather conditions that aid classification in still pictures and videos.

Weather features is described as the atmospheric condition in terms of temperature, wind, cloud and precipitation. These characteristics which are exhibited by atmospheric condition makes weather features highly dynamic in nature causing diversity and lack of discriminate weather features.

The visibility of weather conditions in an image depends on the background scene, illumination of the environment; and the camera intrinsic properties such as exposure time and depth. For example, rain drops characteristics exhibit reflection and refraction of light towards the camera from the surroundings. This results into images or videos motion intensities when dropped at high velocities. The motion intensities rely on the background scene caused by the limited camera exposure. Thus, the size of the rain increases and then decreases with increase in brightness of the environment or surroundings and vice versa (Garg and Nayar 2005).

To address issue pertaining this weather feature, several methods have been employed in the field of computer vision. The works of Derpanis, Lecce, Daniilidis, and Wildes (2012) employed structural information of image processing such as Scale Invariant Feature Transform (SIFT) or Histogram of Oriented Gradient (HOG) which are algorithm based on illumination-invariant features to extract distinct weather features from images/videos. A study conducted by Bossu, Hautière, and Tarel, (2011) also used similar method but applied a mixture model of segmentation technique to separate the foreground from the background to obtain binary image which is used to show the effect of rain or snow in camera images.

Meanwhile, authors such as Lu, Lin, Jia, and Tang (2014) researched on weather features by considering various common weather component (such as sky, shadow, reflection, contrast and haze) that occur every single day while authors, Zhang Z and Ma H,et al, (2016) went further to improve on the work of Lu, Lin, Jia, and Tang (2014) by developing a dictionary which focuses on learning features used to learn and extract only the relevant features required for computer vision and image classification tasks.

Research conducted by Mairal, Bach, and Ponce (2012) was based on image space and transformation matrix. This method builds a dictionary of map sparse features of image patched to intensity values of the output patches. On the other hand, Gao, Tsang, and Chia (2013) improved the effectiveness of this method by employing some implicit features with a focus on mapping of high dimensional features in a sparse coding technique.

In the field of computer vision and machine learning, most researchers have employed supervised method for classification of various weather conditions. In Roser M. and Moosmann F. (June 2008), the authors used Support Vector Machine (SVM) to classify images taken by a driver support system in an open weather of heavy rain or light rain. The datasets of 500,000 images showings expressway were collected from 150 video sequences. The features extracted from each image were minimum brightness, local contrast, hue, saturation and sharpness. The result showed that images of heavy rain have high classification accuracy than images of light rain.

In another situation, Chen, Yang, and Lindner (2012) used SVM to classify multi-class weather feature vectors of sunshiny, cloudy and hazy. The fascinating part of this research work is the area of pre-processing segment that come first before the classification technique. In this technique, the sky portion is isolated from the weather images to prevent conflict with the non-sky features. After obtaining the sky features from the image, Multiple Kernel Learning (MKL) is used to select a subclass of the features from a feature class automatically. The dataset used for this work contains 1,000 images gathered from a specific location.

Likewise, Yan X., Luo Y., and X. Zheng, (2009) researched on alternative method using AdaBoost to classify weather images obtained by mean of vehicular camera. This is an effective ensemble procedure often applied in pattern recognition; used to distinguish between sunny, rainy and cloudy weather conditions. The dataset used for this work was about 2,500 images extracted from videos camera attached to the moving vehicle on the street. The feature vector used composed of brightness, hue, gradient magnitude, saturation, and the average of grayscale values computed from several locations within the area of concerns.

In the work of Elhoseiny, Huang, and Elgammal, (2015), the authors used deep learning techniques to perform weather recognition task by slightly modifying AlexNet as suggested by Krizhevsky, Sutskever, and Hinton (2012). This was achieved using previous trained ImageNet model to classify weather images into sunny and cloudy images. The dataset of 14,000 sunny or cloudy weather images were used to retrain the classifier. The slightly modified AlexNet technique was compared to SVM classifier with same dataset. The result showed that the deep learning outperformed the SVM.

Another deep learning method used to recognized extreme weather conditions (Zhu, Zhuo, and Qu, 2016). The author used GoogleNet architecture of Szegedy, Liu, Jia, (2015) to classify four different weather conditions: sunny, fog, rainstorm and blizzard. Firstly, the author pre-trained the deep learning network on the dataset of ImageNet, after which fine turning was done on it with a previously collected dataset. The dataset used contained 17,000 images showing different complex weather scenes.

## 2.2 ENSEMBLE METHODS

The primary concept of ensemble technique is to combine multiple classifier weights to obtain a better classifier that outperform every individual classifier that makes up the ensemble classifier as illustrated in Figure 2.1 in the work of Hansen and Salamon, (1990).

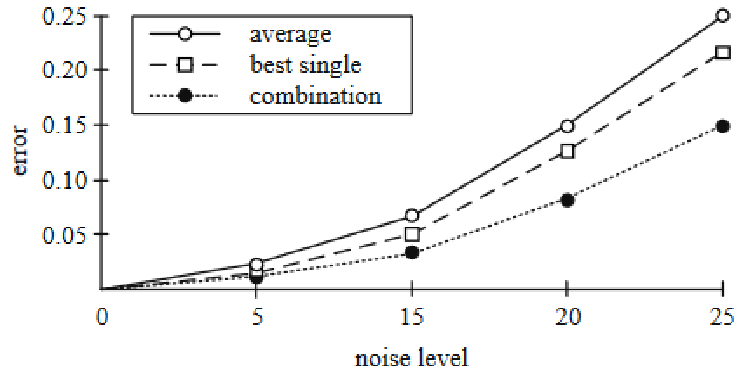



Figure 2.1: Ensemble classifier outperform every constituent classifier
(Source: Hansen and Salamon, 1990)

According to Zhi-Hua and Zhou (2012) ensemble method is defined as committee-based learning or multiple classifier systems

Figure 2.2 depicts the most common ensemble architecture. An ensemble classifier composes of several learners called based leaners (such as logistic regression, Support Vector Machine or any other type of learning algorithms) and a combined learner. The method was originally designed to reduce variance thereby improving the accuracy of the base learners as advanced by Zhang C. and Ma Y., (2012). Most base learning algorithms have similar learning techniques leading to homogenous ensembles whereas when the based learning algorithms have different learning techniques and produces different errors, these are called heterogeneous ensemble.

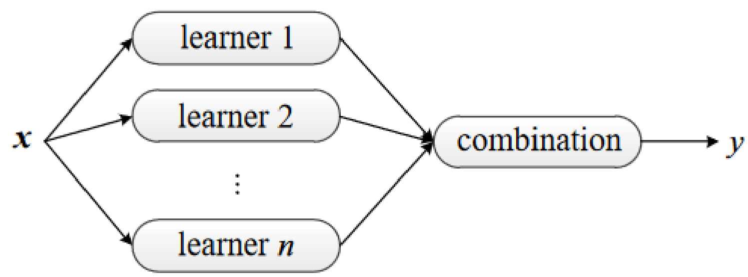



Figure 2.2: Ensemble Architecture (Source: Zhi-Hua Zhou, 2012)

The main purpose is to reduce generalization error in the combined classifier compared to a single classifier. This method strengthens the weak base classifier in the overall system. The combination function of the combined classifier is usually integrated by majority voting for classification task or a weighted average for regression task. For instance, research by Dietterich (2012) explained the three fundamental keys behind the exploitation of ensemble methods which are statistical, computational and representation. More so, the correlational strength method by Breiman L, (2001) and the decomposition of bias-variance technique in the earlier research by Kohavi and Wolpert, (1996) also explain why the ensemble method works.

The bias-variance-covariance decomposition by Breiman L, (2001) has been the major theory behind ensemble methods performance justification over its individual predictions. The keyword in this concept is diversity. The method is applicable to both regression and classification task. The research work of Ren, Suganthan, and Srikanth, (2015) proves that ensemble model generates smaller mean square error when compared to the average square error of the individual models. The average square error is the main cause of ambiguity decomposition in a single dataset. However, with multiple datasets, the bias-variance covariance decomposition was illustrated by Brown and Wyatt, (2005) and Geman, Bienenstock, and Doursat, (1992) and the equations are illustrated from (2.1) to (2.4):

$$E(f-t)^2 = bias^2 \ + \frac{1}{M}\, var + \left(1-\frac{1}{M}\right) covar \qquad (2.1)$$

$$bias = \frac{1}{M} \sum_i E(\{f_i\} - t\,) \qquad (2.2)$$

$$var = \frac{1}{M} \sum_i E(f_i - E\{f_i\})^2 \qquad (2.3)$$

$$covar = \frac{1}{M(M+1)} \sum_i \sum_{j \neq i} E\{(f_i - E\{f_i\})(f_j - E\{f_j\})\} \qquad (2.4)$$

From the equations (2.1) to (2.4), variable $t$ is the unknown target and $f_i$ is the result from each classifier and $M$ is the total number of classifiers. The average bias component measures the average difference between the outcome of the combined classifier and the expected result. The second component is the combined classifier with average variability, and the last component is the covariance of an average pairwise. The generalization error depends on the three properties of the decomposition components which must balance against each other to obtain the best performance. The covariance between individual models will reduce the percentage variance in the overall system, such as increasing the number of models is proportional to increase in covariance while lead to reduction of variance in the overall ensemble method.

The research work of Pisetta (2003), and Zhang, Ren, and Suganthan (2014) resulted into an ensemble technique called bagging method. The method is noted for drastically reducing the variance of the combined classifiers while Breiman L, (2001) and Zhang and Suganthan, (2014) produced another ensemble method that boosts the weak classifiers by reducing bias and variance.

Several research works have been done to prove the validity of using ensemble methods, such works are stochastic discrimination (Domingos, 2000), strength-correlation (Breiman, 2001) and margin theory (Kleinberg, 1990). All these works have shown that they can be alternatives to the decomposition of bias-variance-covariance (BVC) method (Schapire R. E. and Freund Y., 1998).

The accuracy of any ensemble methods primarily relies on the diversity of the individual classifiers that constitute the ensemble method. It is impossible that different classifiers provide same outputs although the inputs are same. Therefore, the error made by individual classifier in an ensemble method can be corrected by another classifier(s). However, there is no standard theory that explains the rate of diversity between the constituent of a combined base model that contributes to the overall accuracy of a meta-classifier of an ensemble method. However, research by Freund and Schapire, (1996) classified the ensemble method diversity into "data diversity, parameter diversity and structural diversity" respectively.

Data Diversity involves partition of the original dataset into multiple sub-dataset to train different classifier. Ensemble methods that use data diversity in their model are AdaBoost (C. Zhang and J. Zhang, 2008), bootstrap aggregation (Ren, Zhang and Suganthan, 2016), random subspace (Breiman,1996 and. Ho, 1998), and Random Forest (Breiman, 1996).

The second group is the parameter diversity which generates different classifier outputs based on the use of different parameter settings. The use of the same training dataset on the same base classifier with different parameter settings may still result in varying output.

The last group is the structural diversity which is induced by having base classifiers with different structures, parameter settings and arrangement. This type of ensemble method is referred to as heterogeneous ensemble (Tan, Li, and Qin, 2008).

Ensemble techniques application have been proven to be very effective in a wide spectrum of real-world problem domains. In an online competition in 2009, Ensemble method was used to improve Netflix[1] prediction by 10% accuracy. This ensemble technique was based on user preferences to predict how a user will enjoy a suggested movie.



[1]http://www.netflixprize.com/

In the field of computer vision, ensemble method has been used for object detection, recognition and tracking. Authors such as Viola and Jones, (2001 and 2004) proposed a framework that combine AdaBoost with a cascade architecture for face detection in 0.067 seconds for a 384 x 288 image;- the findings of the study revealed that this was fifteen times faster than the best face detectors, while detection accuracy was almost similar.

Another important role of the ensemble method in computer vision is pose-invariant face recognition (Huang, Zhou, Zhang, Chen 2000), particularly for identifying face with different varying degree of rotations. The main concept is to combine several neural networks specific views with a unique crafted module. This method outperforms conventional techniques by not requesting for pose information as input as compared with normal conventional method, instead with output pose-information. A similar technique was later employed by Li et al, (2002) for multi-view face detection.

For object tracking, Avidan, (2007) worked on ensemble tracking, which is an online ensemble classifier that differentiate between object and background. This method updates weak classifier constantly by adding or removing classifiers at any time. This method injects new information about the transformation in the background and the object appearance. This work demonstrates that ensemble tracking framework is highly efficient within a few frames per second without tuning into a variety of online video applications.

Furthermore, Corona, Giacinto, Mazzariello, Roli and Sansone, (2009) showed that ensemble method can be useful in computer security problems. Reason being multiple abstraction levels can be used to monitor each activity performed on computer systems, while the important information could be collected from multiple information sources.

Ensemble method was also employed to detect intrusions (Giacinto, Roli, and Didaci 2003). The proposed method considered different types of input features and these were fed into different base learners and their combined outputs were used to make the final decision. Five-year laters, Giacinto , Perdisci, Rio and Roli (2008) built upon the previous work to develop a framework which can detect intrusion that has never been seen before.

In computer aided medical diagnosis, ensemble method can increase the rate of reliability of diagnosis. For instance, Zhou, Jiang, Yang, Chen (2002a) developed lung cancer cell identification using a two-layered ensemble architecture. The first layer dealt with mid cases and the prediction was based if only all the base learners agree; otherwise the case would move to the next layer to make for further analysis on other cancer cases fed as input. The second layer was designed to differentiate between cancer types. This method recorded high true-positive rate with a low false-negative recognition rate.

Linking with the previous paragraph, a study by Polikar, Topalis, Parikh, Green, Frymiare, Kounios, Clair (2008) furthered the work of Zhou, Jiang, Yang, Chen (2002a) for early diagnosis of Alzheimer's disease by considering multiple data EEG (electroencephalogram) channel as against a single channel used by Zhou, Jiang, Yang, Chen (2002a). Each data source obtained from different electrodes, different stimuli response, and different frequency bands are trained by different base learners, and the final diagnosis result is based on the combined output from various sources.

In addition to the previous mentioned application of ensemble methods, these were also used in other domains such as detection of credit card fraud (Chan, Fan., Prodromidis , and Stolfo ,1999 and Panigrahi, Kundu, Sural, and Majumdar, 2009), fault diagnosis in aircraft engine (Goebel, Krok, Sutherland 2000, and Yan & Xue, 2008), bankruptcy prediction (West, Dellana., and Qian 2005),species distributions forecasting (Ara ́ujo and New, 2007), forecasting of electric load system (Taylor and Buizza, 2002), artist and genre of music classification (Bergstra, Casagrande, Erhan, Eck, and Kegl., 2006), weather forecast (Maqsood et al., 2004; and Gneiting and Raftery, 2005), and classification of protein structure (Tan et al., 2003, Shen and Chou, 2006).

In the next section, the discussion focuses more on key algorithms concept behind the construction of any ensemble methods.

### 2.2.1 BOOTING

The booting algorithm is an algorithm that can convert weak learners to strong learners. The first booting algorithm was introduced by Schapire, (1990) to answer an important question posed by Kearns and Valiant, (1989) on whether problems for weak base learner and strong base learners are equal.

Five years later, Freund and Schapire (1996) proposed the AdaBoost algorithm. The main principle behind boosting algorithms is that it can correct the mistake made by a weak classifier. To achieve this, equal weight is assigned to each training set at the beginning, but in each iterative step, the weights of all incorrect classifiers will increase while the weights of correct classifiers reduce. As a result, the weaker base learner is compelled to focus on the incorrect data in the training set. By the end of the iteration, the classifiers are expected to complement one another.

Consider binary classification on class labeled as {-1, +1}, the algorithms assume training set consisting of m examples. The classification for the unseen data is made by voting on all the base learners or classifier $\{C_t\}$, each having a weight of $\alpha_t$. This is expressed mathematically as:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t \cdot C_t(x)) \quad (2.5)$$

The second version of AdaBoost algorithm described by Freund and Schapire, (1996) perform similarly in the way or method in which binary classification task were done. However, if there are differences in multiclass classification problems, and it is expressed by the equation (2.6) as:

$$H(x) = \underset{y \in dom(y)}{\text{argmax}} \, ( \sum_{t:C_t(x)=y} \log \frac{1}{\beta_t}) \quad (2.6)$$

The performance of boosting algorithms appears to increase for two main reasons:

1. The meta or combine learner error on the training set is smaller when compared to individual base learners; and
2. Likewise, the variance of the meta or combined leaner is lower than the individual base leaners.

However, boosting has its own drawback as it is prone to overfitting according to Quinlan, (1996). A proposed solution to the overfitting challenges of boosting algorithms is to maintain a small number of iterations as possible.

### 2.2.2 BAGGING

Bagging algorithm is a bootstrap and aggregation algorithm Breiman, (1996) that combines independent base leaners whose output errors are reduced drastically. For example, Consider N samples size of a random generated training dataset, drawn with replacement. The output result may contain some subset of training set that is repeated multiple times while others may be left out. This algorithm utilizes the bootstrap distribution techniques for generating different bases leaners (Efron and Tibshi-rani, 1993).

To compensate for these repeated errors, Bagging algorithms employs popular aggregating strategies for the first learning algorithm output, that is, voting techniques for classification task, while the averaging method for regression. For example, to predict an unseen input in classification task, the input is fed into the base leaners, and the base learners' output labels are collected, and voted for. The winning label becomes the final prediction.

Random Forest algorithms is an example of creative version of bagging that implements the research technique based on ensemble of trained decision trees (Kirchne et al, 2010). For an example, random forest can perform random selections of features subset as described by the works of Riddick, G. and Song, (2011) in random subspace models.

The works of Breiman (1996) proved that efficiency of bagging method on "erratic" learning algorithms when little changes in the training dataset result in large changes in final prediction. The out-of-bag examples method is used to measure the goodness of bagging base leaners, after which the general error caused by the ensemble method can be predicted.

## 2.3 COMBINNING ALGORITHMS

The method of combining different learning algorithms may be divided into two main categories: combined base learning generally suitable for solving problems whose individual base learners do similar function but have different success rate. Nevertheless, such algorithms are prone to outliers' vulnerabilities and to erratic execution of algorithms. Whereas the meta-learning algorithms are more powerful but subtle to all the problems associated with the poor learning such as long training time, and over-fitting.

The simple combining methods are explained below:

- **Uniform Voting**: This method involves each base learning algorithms having equal weight. The prediction of the unlabeled input is performed by obtaining the class with the winning number of votes. This can be expressed mathematically as shown in equation (2.7) as:

$$Class(x) = \underset{c_i \in dom(y)}{\text{argmax}} \sum_{\forall k c_i = \underset{c_j \in dom(y)}{\text{argmax}} \hat{P}_{M_k}(y=c_j|x)} \tag{2.7}$$

  Where $M_k$ is k learning algorithm, while $P_{Mk}$ (y = c|x) is the probability of y is equal to chance of obtaining the value c given a positive input x.

- **Bayesian Combination**: The Bayesian combination was proposed by Buntine (1990). The concept involves associating weight to each learning algorithm as the probability of the learning algorithm given a data set S.

$$Class(x) = \underset{c_i \in dom(y)}{\text{argmax}} \sum_k P(M_k|S) \cdot \hat{P}_{M_k}(y = c_i|x) \tag{2.8}$$

  Also, $M_k$ is the learning algorithms of probability P ($M_k$ | S) given the training dataset S. The probability of P ($M_k$| S) relies on the learning algorithm's outcomes, that is, $M_k$.

- **Naıve Bayes Method:** This method extends Naive Bayes rule for combining one or more learning algorithms as illustrated in equation (2.9):

$$class(x) = \underset{\substack{c_j \in dom(y) \\ \hat{P}(y=c_j)>0}}{\operatorname{argmax}} \hat{P}(y=c_j) \cdot \prod_{k=1} \frac{\hat{P}_{M_k}(y=c_j\,|x)}{\hat{P}(y=c_j)} \tag{2.9}$$

- **Entropy Weighting:** The main technique behind this combination method is to apportion weight to each learning algorithms which is inversely proportional to the entropy of its vector classification as shown in equation (2.10) to (2.11):

$$Class(x) = \underset{c_i \in dom(y)}{\operatorname{argmax}} \sum_{k:c_i = \underset{c_j \in dom(y)}{\operatorname{argmax}} \hat{P}_{M_k}(y=c_j|x)} Ent(M_k, x) \tag{2.10}$$

Where:

$$Ent(M_k, x) = -\sum_{c_j \in dom(y)} \hat{P}_{M_k}(y=c_j\,|x) \log\left(\hat{P}_{M_k}(y=c_j\,|x)\right) \tag{2.11}$$

- **Density-based Weighting**: This method used various trained learning algorithm dataset obtained from different sources to assign weights to the learning algorithms. Mathematically, it is written as:

$$Class(x) = \underset{c_i \in dom(y)}{\operatorname{argmax}} \sum_{k:c_i = \underset{c_j \in dom(y)}{\operatorname{argmax}} \hat{P}_{M_k}(y=c_j|x)} \hat{P}_{M_k}(x) \tag{2.12}$$

### 2.3.1 BENEFITS OF COMBINNING LEARNING ALOGRITHMS

Following the generation of ensemble base learners, ensemble methods try to find a way to combine the best base learners to accomplish a strong generalization capability. This combination performs an essential role in ensemble method. The works of Dietterich (2000) highlighted three fundamental benefits why the combination of ensemble method is so important. These are:

- **Statistical issues**: When available hypothesis is too large to explore for inadequacy of training data, there may be several subsets of the available hypothesis might give the same result as the training dataset. Therefore, there is risk that the trained dataset chosen might not be able to predict the future of the unknown data set. Conversely, combining different available dataset reduces the risk of selecting the wrong hypothesis.

- **Computation issue:** Learning algorithm often get stuck at the point of local optima, that is, finding the best hypothesis can be difficult with enough dataset. The solution is to run different hypothesis at different local search points from different starting points to reduce the risk of selecting an incorrect local minimum value.

- **Representational Issue**: Representing unknown hypothesis in most machine learning algorithms is difficult as representing in the hypothesis space. Therefore, combining hypothesis might lead to expansion of space representation that learning algorithms might use to form a more precise estimate of the true unknown hypothesis.

In summary, the highlighted issues in section 2.3.1 explain why most traditional learning algorithms fails High "variance" issue is suffered by learning algorithms as a result of statistical issues, while high computational "variance" is as a result of computational issues, whereas high "bias" in learning algorithms is caused by representational issue. Hence, combining various learning algorithms, reduces variance as well as bias of learning algorithms (Xu et al., 1992, Bauer et al, 1999, Opitz et al, 1999).

## 2.4 STACKED ENSEMBLE METHOD:

The works of Wolpert (1992), Breiman (1996), and Smyth and Wolpert (1998) shows that stacking is a technique where a combined learner is trained to combine different base learners. The base learning algorithms are referred to the first-level learners, while the combined learner is termed meta-learner or second-level learner.

The main concept is to use original training dataset to train the first level learners, these first level learners generates a new data set which is termed the new input features vector. The new features vector is mapped to the original data labels. These new features vector is used for training the meta-learner or second-level learner. Combining different first-level learners using different learning algorithms is what is called stacked ensembles. These are often heterogeneous in nature, though construction of homogeneous stacked ensembles is possible.

In another perspective, stacking is viewed as a generalize framework for many ensemble methods while in another manner as a specific combination of different learning algorithms. The author Breiman (1996), proves the success of stacked regression. He applied different sizes of regression trees as the first-level learners, that is, learners with different variables, and the meta-learner's output are based on non-negative co-efficient of least-square linear regression model. The non-negative constraint applied was found to be important to ensure the stacked ensemble method outperformed every individual first-level learner.

In the classification task of stacked ensemble method of Wolpert (1992) proves that for any stacking algorithms, the first-level learners' selection and the type of features vector produced for second-base learner are important.

The authors Ting and Witten, (1999) suggested that class probability should be used as replacement for class label as features since this considers not only the predictions but also the confidence of the based learners. The authors also suggested the use of multi-response linear regression (MLR) as the second-level learning algorithm, which is also a type of the least square linear regression algorithm. In Seewald (2002), the author recommended that in MLR, diverse sets of features should be use for linear regression problems.

In 2003, stacking methods was compared to Bayesian Model Averaging (BMA) by Clarke, (2003). This method assigned different weights to different models based on posterior probabilities. The experimental results show that stacking method is more accurate than BMA, because BMA is subtle to model approximation error.

## 2.5 LITERATURE RESEARCH GAP:

Even though several research works have been done on weather classification, multi-class weather classification is still a difficult task to perform due to diversity and lack of distinct weather characteristic or features. Most of the researches were based on clear weather assumption which are high affected by the dynamic nature of weather. (Nashashibi, Charette, and Lia 2010, Kurihata, Takahashi, Ide, Mekada, Murase, Tamatsu and Miyahara, 1963, Woo, Jung, Kim, and Seo, 2010) ,Katsura, Miura, Hild, and Shirai 2003). The bad weather conditions usually affect the quality of images or videos, and it is necessary to use the weather information to correct the image/video processing algorithms to achieve better performance.

Weather classification from still images is usually time consuming, difficult and requires experience skills and knowledge to identify distinct weather features. The extant literature indicates that researchers have in the past used multiple kernel learning (MKL) to determine the optimal weather features combination for weather classification. This method discards features that are weak to contribute to the overall recognition.

As of the time of writing this project, no stacked ensemble technique has been attempted to solve this issue. Hence, this research work addresses the problem of weather features combination and recognition using the state-of-art ensemble methods

# CHAPTER THREE
# RESEARCH METHODOLOGY

## 3.0 CHAPTER OVERVIEW

As stated in Chapter One, the aim of this research was to develop a novel framework that can differentiate between different weather images (such as rain, cloudy, sunrise and sunset). In particular, the focus of this research was analysing clear weather images. However, this task is challenging not only because of discriminate weather features in images but also because of a lack or few clear weather datasets with the required characteristics.

In this chapter, Section 3.1 explains the experimental tools and configuration used to build the model while in Section 3.2 a description of how the training dataset for the model was acquired is presented. In Section 3.3, the proposed ensemble model and its training are described. Whereas the last Sections 3.4 explains how the research approach to diversity of stacked ensemble model was adopted.

## 3.1 EXPERIMENTAL TOOLS AND CONFIGURATION SETUP

This section explains the experimental tools and configuration used for the framework development. Each subsection explains the choice of the framework for the evaluated system and provides background information.

Python programming language is the language of choice in this research work because of its large growing ecosystem for machine learning development. Furthermore, it was chosen because it is a multi-purpose programming language that can be used for quick prototyping in research and development, and in production or commercial environment. The sections below discussed the python  library used and these are as followings:

### 3.1.1 PILLOW

Pillow popularly known as python image library (PIL) is an open source library for image processing. It is used for opening, manipulation and saving of image. In this dissertation, this library was used in conjunction with NumPy library for image processing.

### 3.1.2 SCIPY

SciPy is a mathematics, science and engineering python library that is needed for machine learning tasks. It is an add-on to Python repository. The ecosystem of SciPy is composed of different core modules that are important to machine learning development as highlighted below:

- NumPy: The building block for SciPy that is used to work efficiently with array data;
- Matplotlib: Python core module that is used to plot different kinds of graph and charts from data; and
- Pandas: Is a tool for data structure and manipulation in order to perform repetitive task.

### 3.1.3 SCIKIT-LEARN

The Scikit-learn is a python developed through an open source library that can be used for rapid development and machine learning practice. The library is built or developed upon SciPy ecosystem. Therefore, it is required during scikit-learn installation. The name Scikit is an abbreviation for SciPy toolkit. The library focus is machine learning algorithms for achieving the task of supervised and unsupervised learning. The library also provides API interface for related tasks such as data pre-processing, machine learning algorithms evaluation, and parameter optimisation.

### 3.1.4 PICKLE

Pickle is a python module that can take practically any python object and translate it to byte stream of string serialization. This concept is called "pickling" while the inverse operation of converting back to python object is called "unpickling". This module was used in this dissertation to store weather features for later re-use.

### 3.1.5 SYSTEM CONFIGURATION

The python programme and its machine learning ecosystem were installed on laptop with the following configuration:

- Lenovo Core i5;
- 1.8GHz CPU frequency;
- 8GB system memory (RAM);
- Hard-disk size of 512GB; and
- Ubuntu 13.04 operating system (OS).

## 3.2 MULTI-CLASS WEATHER DATASET (MWD)

To train a meta-learner of a stacked ensemble algorithms for the task of classifying different weather conditions, an unobstructed weather recognition dataset had to be acquired first. Although there are a few or no unobstructed weather dataset that are freely available, none met the exact requirements described in the paragraph below. Therefore, a new set of weather images had to be collected manually online.

### 3.2.1 DESIGN OF THE DATASET

There were several requirements for the images that would be used as stimuli for the data collection. Firstly, for the purpose of easy training and evaluation of stacked ensemble method that can distinguish between different weather conditions, the images had to meet some technical requirements namely:

1. Images should be an outdoor weather image;
2. The weather images should capture some portion of the sky;
3. In order to have a good generalized model, the dataset should contain different distributions of same image scene that is exposed to different weather conditions;
4. After this, we identified the images that meets the exact requirement, we manually annotate the images; and
5. To this end, the collected dataset was organized into different categories required for classification task.

For any outdoor weather image identification task, the sky is the most important weather feature because dynamic characteristics of different weather conditions are exhibited in the sky. On a sunny day, the sky appears to be blue in colour due to scattering of sunlight molecules as it passes through the atmosphere. On the other hand, a cloudy day exhibit most of weather dynamic nature which are defined by different intensity degree of opaque cloud coverage (OCC). Mostly sunny



[2] *http://www.weather.gov/media/pah/ServiceGuide/A-forecast.pdf*

and partly cloudy weather condition are defined to be between 25% and 50% OCC, partly sunny and mostly cloudy are between 51% and 87% OCC. Meanwhile overcast is at 88% OCC and above

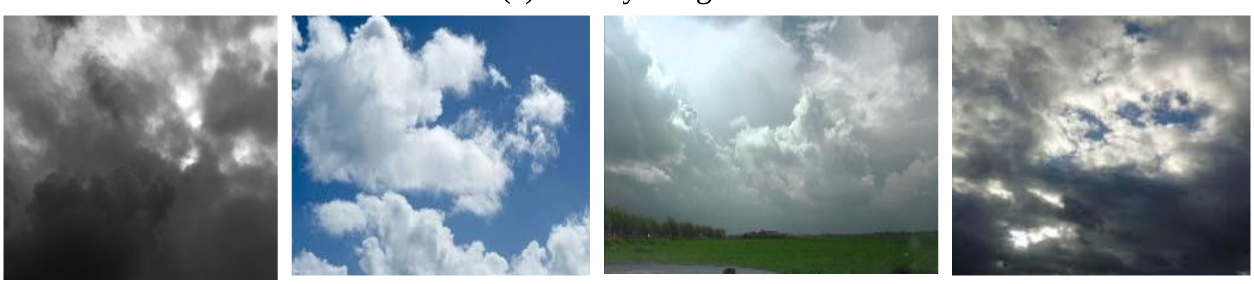
### 3.2.2 DATA COLLECTION

Because there are few or no unobstructed weather recognition dataset, the research approaches multi-class weather recognition by first constructing dataset of unobstructed images of different weather conditions collected from internet sources such as google images, flickre, gettyimages, yahoo images. A total of 1125 weather images were manually collected and annotated as cloudy, sun rise, rainy and sunshine. The partial samples of the images collected are shown in Figure 3.1 and the statistically distribution of Multi-class weather (MWD) dataset is shown in Table 3.1.

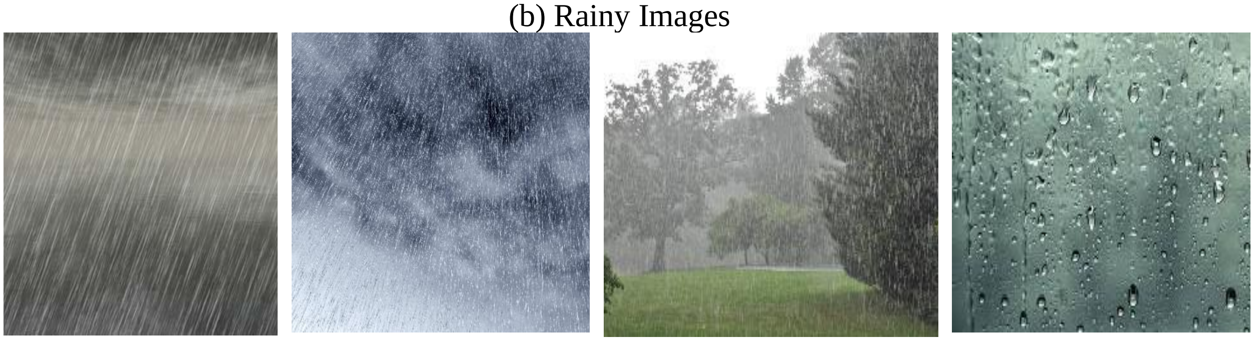
*Table 3.1: The statistically distribution of MWD dataset*

| | ***Cloudy*** | ***Sunshine*** | ***Rainy*** | ***Sunrise*** |
|---|---|---|---|---|
| Number | 300 | 235 | 215 | 357 |

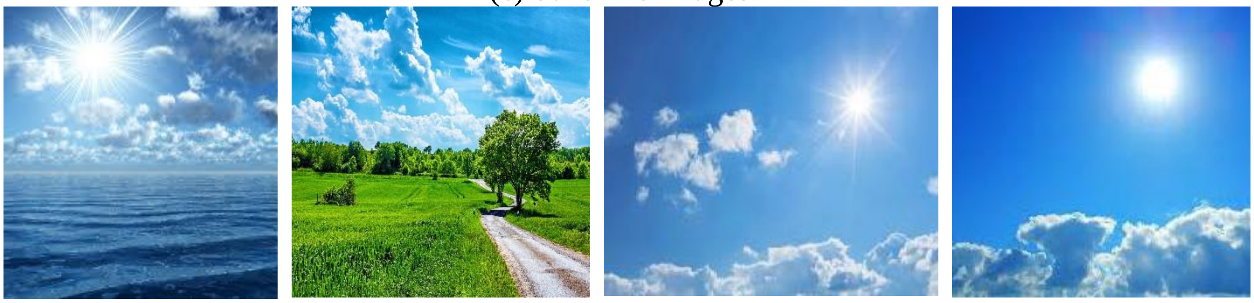
(a) Cloudy Images



(b) Rainy Images



(c) Sunshine Images



(d) Sunrise Images

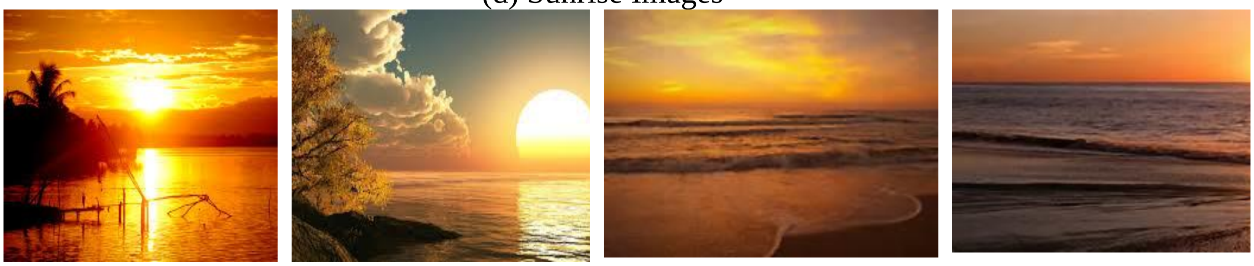


Figure 3.1: Sample of Multi-class Image Dataset

### 3.2.3 MULTI-CLASS WEATHER IMAGE PRE-PROCESSING

In this research, before developing the model, the weather multi-class dataset obtained from different internet sources were carefully studied and analysed. The images were observed so that researcher would be in a position to have different properties such as: aspect ratio, intensity, dimensions. As a result, this prompted pre-processing stage of the experiment. The pre-processing step performed is as explained in the next section.

#### 3.2.3.1 IMAGE SCALING AND ASPECT RATIO

One major constraint in building the learning model or algorithm for the task ahead was the need to resize the different sizes of the dataset images to uniform dimension of 300 by 245px while maintaining the aspect ratio of the images (i.e. the ratio of the height to the width of the image).

Ignoring the aspect ratio could lead to distortion and compression of images. In this research, the pre-processing was performed using the open source python library called Pillow described in section 3.1.1. This was used for reading all the images in the dataset directory, resizing and saving to the new required dimension into a new directory which becomes the new dataset.

Generally, image pre-processing is essential to speed-up training and improve image features by removing unwanted falsification.

## 3.3 EXPERIMENTAL ENSEMBLE METHOD

In the review of related literature as presented in chapter 2, discussion of the effectiveness of ensemble method for solving different problems in different domains was outlined. In this section, we present the experiment overview of employed in this dissertation to solve the challenge of weather classification.

The research introduces a new ensemble framework for identifying different weather scenes from single images as shown in Figure 3.2. Firstly, the framework involves extraction of multiple weather features from each image.

The histogram features are Hue, Saturation and Value (HSV), Gradient, Contrast and Local Binary Pattern (LBP). Therefore, each of the feature vector dataset be denoted as $f1$, $f2$ ,$\ldots\ldots\ldots fn$ having an instance space $x \in X(x1,x2,\ldots\ldots\ldots xn)$ and class label y $\in Y(y1,y2,\ldots\ldots\ldots yn)$ . The total dataset $D$ can be express as {(($x1$, $y1$), ($x2$, $y2$) …... ($xn$, $yn$)}.

Model combination and diversity for ensemble learning was based on SAID (Selection based on Accuracy Intuition and Diversity). The SAID resulted into two stacked ensemble learning algorithms with each of the features $f$ being learned by each of the stacked ensemble algorithms ($L$

= $L1 + L2 + \cdots ..., + Lm$). In this case, the length (m) of the learning algorithm is equal to three (3) in both experiments.

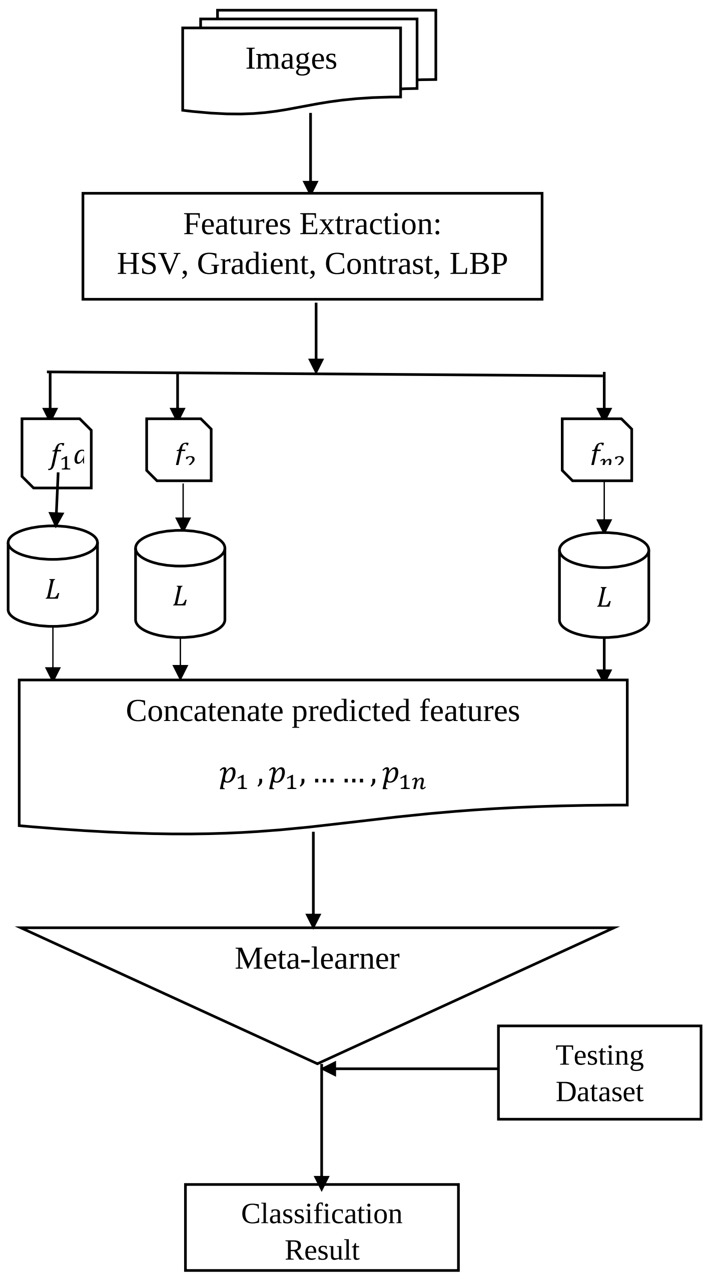



Figure 3.2: Flowchart of our proposed stacked ensemble

The common non-linear algorithm which was considered for model selection is Random Forest Classifier, KNN, Radial base kernel function Support vector machine (RBF-SVM) and the Native Bayes method. This method is used to produce heterogeneous features which form the meta-dataset for the meta-classifier. The following section describes each step-in detail.

For base learner for stacking method, the following four base learners were chosen because they exhibited different variety of biases. The base learns are:

1. K-Nearest Neighbours: This is a clustering algorithm used in unsupervised learning for classification and regression problems;
2. Support Vector Machine using radial base kernel: A supervised learning algorithms also used for classification and regression problems;
3. Naive Bayes: Also, a supervised learning algorithm that is based on probability theory used for classification task; and
4. Random Forest: This is a variant of ensemble method that is based on bootstrap aggregation or bagging.

### 3.4 DESIGN OF DIVERSITY METHOD

Diversity is an important factor that is linked to the success of any ensemble learning scheme. For the purpose of this study, diversity was considered at the based learners' level to determine how the base learners were combined. This is based on training base learners on copies of the same multi-class weather training data set. The diversity method ensures that the optimal-hyper-parameter (i.e. model selection) is selected for learning algorithms using a quantitative approach, which leads to the optimal performance of the individual base learners. The quantitative approach is based on experimental accuracy of base learners which are coined and referred to as Selection

Base on Intuition, Accuracy and Diversity (SAID). SAID uses a minimum 10% variance to between base learners to determine how base learners will be combined.

## 3.5 FEATURE EXTRACTION AND FEATURE SELECTION

For any successful pattern recognition problems, selecting the right features or interest point from images is very important particularly in distinguishing images of the same scene. Unfortunately, expressing weather features taken from the same scene under different weather conditions requires analysing several low-level image features.

The general weather features involve extracting the characteristics exhibited by most weather condition which textures, colour and shape. For the purposes of this study, different python functions were written to extract 128 dimensions local binary pattern (LBP), 384 dimensions of Hue, Saturated and Value (HSV) (i.e. 128 dimensions of histogram of H, 128 dimensions of histogram of S and 128 dimensions of histogram of V), 128-dimensions of gradient magnitude and the 128 dimensions contrast features was computed using the equation 3.1 to form robust feature vectors.

$$C = (I_{max} - I_{min}) \frac{1}{(I_{max} + I_{min})} \tag{3.1}$$

where $I_{max}$ and $I_{min}$ are the intensities maximum and minimum value for each image in the dataset. Each of the extracted feature were saved with python pickle library described in section 3.1.4

Heterogeneous ensemble selection method was also used. This method involved the use of different base selection methods to train different extracted weather features. The predicted output from each selector is a feature subset or a feature ranking. To obtain the final output feature, it was imperative to combine all the weather feature subset obtained from the base selectors. This new feature from the base selectors is also saved for re-use with python pickle library.

### 3.6 MODEL SELECTION

Diversity is an important factor for obtaining accuracy of an ensemble method. Classifiers' diversity leads to unrelated classification, which in turn improves classifiers performance. However, there is no standard theory that explains how individual models' diversity contributes to the overall ensemble method performance. Therefore, classifiers selection is based on intuition, accuracy and diversity criteria. This method in the research under study is referred to as SAID. Based on intuitions, the selection was bench marked to a minimum of 10% accuracy difference between models.

### 3.7 CONCLUSION

This chapter highlighted the step-by-step procedures that chart the course of this research experiments in the following manner:

1. Python is the programming language of choice for this research;
2. Thus, the reasons for the use of different libraries and framework were explained;
3. The research flowchart diagram of the methodology was also presented. This served as overall guide for each step taken along the experiment journey;
4. The Multi-class Weather Dataset (MWD) criteria for weather image collection and categorization were presented;
5. Thereafter, MWD were pre-processed to aid consistency in the dataset and to reduce experiment computation time.

6. From the MWD, different weather features were extracted and store in pickle format for later use in the experiment.
7. Finally, the model selection technique which was based on SAID was employed for construction of the stacked ensemble method.

# CHAPTER FOUR

# RESULTS AND DISCUSSION

## 4.0 CHAPTER OVERVIEW

This chapter discussed the experimental results that were done to find the best ensemble architecture and the relevant parameters for constructing heterogenous stacking models to achieve the research goals described in section 1.4. Hence, Section 4.1 explains the results that governs the construction of the stacked ensemble algorithms or models, while Section 4.2 and 4.3 describe in detail how the results obtained for extracted features contributed to the overall identification of different still weather images. The final section compares the result of the meta-learners with the results of the individual algorithms that makes up of the stacked heterogenous model.

## 4.1 RESULT OF BASE LEARNERS' PERFORMANCE

To construct a good performing stacked ensemble method, the base leaners must possess high diversity that leads to unrelated classification output. The experimental understanding of these features' contribution to the overall performance or accuracy of stacked/combined ensemble algorithm were based on the proposed SAID concept described in Section 3.6.

To apply SAID technique, the combined weather features saved in pickle format in Section 3.5 were extracted and trained with the four (4) different potential classifiers or base learning models which are K-nearest neighbourhood (KNN), Radial base kernel function Support vector machine (RBF-SVM), Native Bayes and Random Forest. This experiment was performed using the Sklearn library functions described in section 3.1 using cross validation fold setting of five (5).

Using the percentage mean accuracy as metric for performance measurement, the results of these base learners are shown in Table 4.1 and Figure 4.1

Figure 4.1: Base learner results

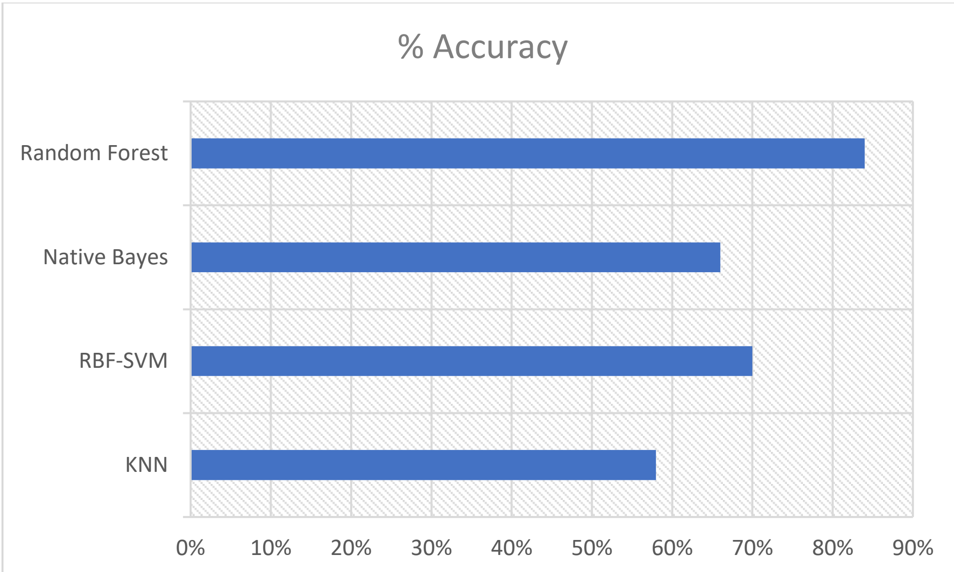



Table 4.1: Base learner results

| Method | % Mean Accuracy |
|---|---|
| KNN | 58% (+/- 0.02) |
| RBF-SVM | 70% (+/- 0.01) |
| Native Bayes | 66% (+/- 0.02) |
| Random Forest | 84% (+/- 0.02) |

Referencing both Table 4.1 and Figure 4.1, the result shows that the Random Forest and RBF-SVM were to the two best base models that contributed more while KNN contributed less to the overall performance of the meta-learner accuracy when used with MWD.

The construction of the stacked models for the research experiments were predicate on SAID concept discussed in Section 3.4. This SAID concept led to construction of two stacked models experiment for weather recognition, namely:

- Experiment I: The base learners that constitute the stacked model are KNN, RBF-SVM and Random Forest with Gradient Boost as the meta-learner.
- Experiment II: The second experiment-based learners for the stacked model are on KNN, Native Bayes and Random Forest. Also, with Gradient Boost as the meta-learner.

## 4.2. RESULT AND DISCUSSION ON FEATURES PERFORMANCE

The performance of any computer vision features plays an important role to the success of any classification task. Hence, in Section 3.5, different features that were extracted from multi-class weather dataset and stored in pickle format were unpickled and trained with the two constructed stacked models described in Section 4.1. Using the percentage mean metric as unit measurement. The experimental results of each feature performance contribution to the overall recognition rate are shown in the Figure 4.2 and Figure 4.3 respectively.

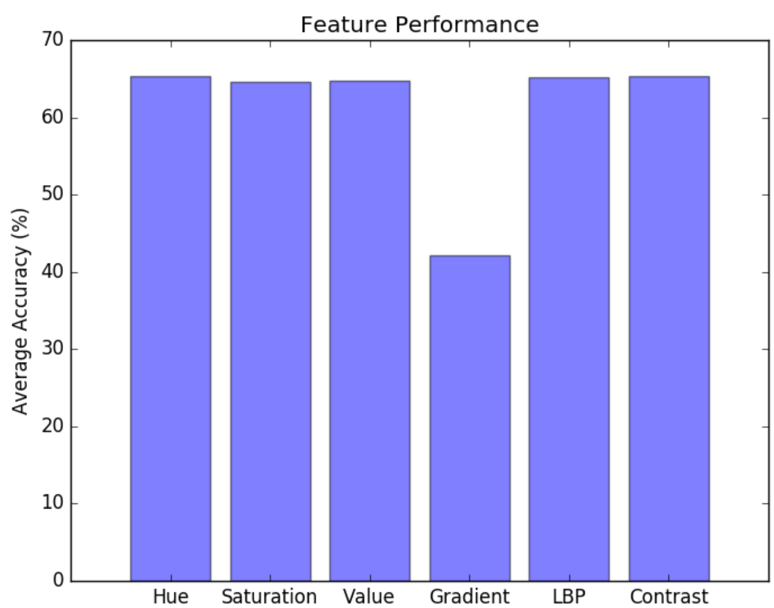



Figure 4.2. Experiment I - Percentage Feature Performance

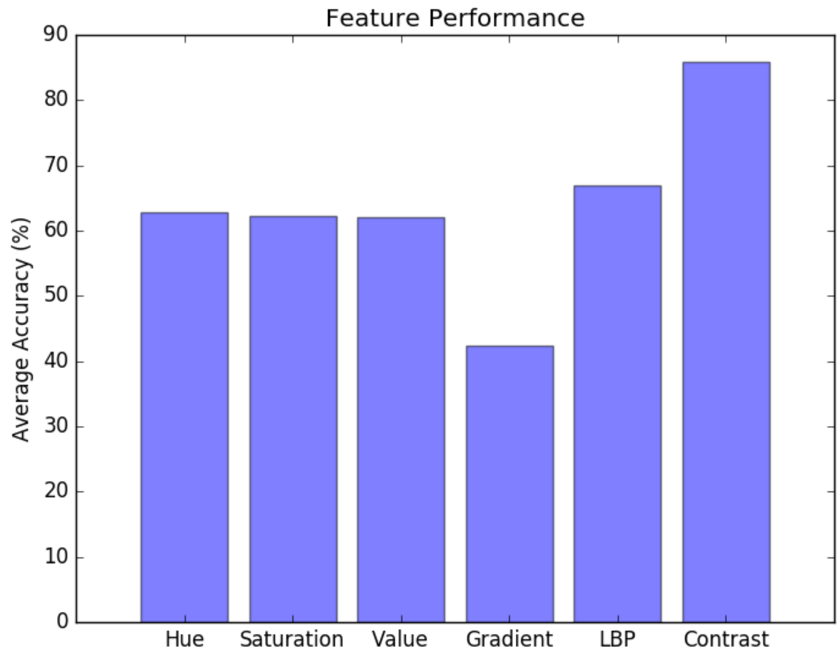



Figure 4.3. Experiment II - Percentage Feature Performance.

The Figure 4.2, which represented the experiment I, had most of the extracted weather features contributed over 60% to the overall performance of the meta-classifier except for the gradient feature whose result of performance is less than 50% contribution to the overall experiment performance.

In the same vein, Figure 4.3 is the feature results graph that illustrates experiment II. This again clearly shows the poor performance of gradient feature when compared to other weather features that performed above 60%. It also noted that the contrast feature performed significantly well in this experiment, with contribution over 85%.

### 4.3 RESULT AND DISCUSSION ON WEATHER STILL IMAGES

This section assesses the performance of the multi-class weather identification framework using two different stacked ensemble model combinations described in section 4.1 above. The diversity of the experiment was based on SAID. In both experiments, the cross-validation technique was used with random selection of 80% of MWD. The number fold is set to value of five (5) to be used by the stacked classifiers for dataset training and validation of each feature and the remaining dataset were used for testing the meta-classifier as illustrated in Figure 3.2. For the stacked classifier, the experiment parameters use the default Sklearn ensemble setting except for the number of estimators of Random Forest Classifier and Gradient Boot Classifier that were set to one hundred (100). The outputs of each stacked classifier produce heterogeneous features which are combined to form new dataset for the metaclassifier.

The new dataset is used for training and validating the Gradient Boot meta-classifier before being tested by unseen dataset. To make the experimental result convincing, each experiment was repeated ten times and the percentage mean results for each experiment were recorded and tabulated as shown in table 4.2 and 4.3

Table 4.2: Mean classification accuracy of SAID experiment I with MWD

| Dataset | Mean Accuracy (+/- Std) |
|---|---|
| Cloudy | 81.70% (+/- 2.21) |
| Rainy | 93.80% (+/- 2.57) |
| Sunshine | 88.20% (+/- 3.43) |
| Sunrise | 83.10% (+/- 3.70) |

Table 4.3: Mean classification accuracy of SAID experiment II with MWD

| Dataset | Mean Accuracy (+/- Std) |
| --- | --- |
| Cloudy | 81.50% (+/- 3.37) |
| Rainy | 95.20% (+/- 1.69) |
| Sunshine | 88.40% (+/- 2.12) |
| Sunrise | 81.70% (+/- 2.16) |

Table 4.2 shows the result of experiment I with average recognition rate of each still images of MWD greater than 80%. The rain still images were the most recognised images in the dataset.

In the same manner, experiment II results illustrated in Table 4.3 also shows over 80% recognition rate for each of the still image of MWD. In both experiments, the rainy still images have the highest rate of recognition accuracy.

## 4.4 RESULT AND DISCUSSION ON ALGORITHM COMPARISON

In this section, the experiment validates the hypothesis that states that the meta-classifier or model yield better performance than the existing model by taking advantages of the weakness of its existence as described in section 2.4. Therefore, the stacked ensemble learning model based on SAID diversity performance were measured against its base learning models. The average percentage performances are shown in the Table 4.4:

Table 4.4: Comparison of base learners with Ensemble Method based on SAID method

| Method | Mean (+/- Std) |
| --- | --- |
| KNN | 58% (+/- 0.02) |
| RBF-SVM | 68% (+/- 0.01) |
| Native Bayes | 66% (+/- 0.02) |
| Random Forest | 80% (+/- 0.02) |
| SAID Experiment I | 85% (+/- 0.02) |
| SAID Experiment II | 86% (+/- 0.02) |

The result of Table 4.4 shows that meta-learner of both experiments performed better than the base learners that made-up stacked ensemble method. This position was also supported by previous research works (Wolpert 1992, Breiman 1996, and Smyth and Wolpert 1998).

## 4.5 CHAPTER SUMMARY

The result of this study shows that, from the base model of the ensemble methods, the performance of the features was measured using percentage accuracy of each features in both experiment. Furthermore, the findings as highlighted in Figure 4.2 shows the percentage features performance for experiments based on SAID method I. Moreover, it can be observed that most of the features contributed to the overall classification of the gradient boost meta-classifier while Figure 4.4 shows the percentage features performance for SAID method II. The memrging result also revealed that contrast and LBP performance contribute more towards the classification accuracy than the performance accuracy of method I while HSV perform equally well in both methods. However, the performance contribution of gradient magnitude to both method is low. It can be observed that SAID method II performs better than SAID method I based on large range of diversity. Meanwhile, Table 4.2 and Table 4.3 shows the average classification result of method I and II with MWD with cross-validation of 10 respectively. In both experiments, the result showed that rainy images have the highest average classification percentage of 93.80% in experiment I and 95.20% in experiment II respectively. On the other hand, cloudy images revealed 88.20% in SAID method I and 88.40% in SAID method II. For sunrise, 83.10% and 81.70% average classification accuracy was achieved in method I and method II respectively, while in 81.70% and 81.50% average classification of cloudy images were correctly classified. Lastly, The base learning algorithm performance was compared to their combined algorithm based on SAID diversity techinique. In Table 4.4, the results of the study shows that the combined algorithm based on SAID technique outperforming its constituent base learners.

## 4.6 CONCLUSION

This chapter explains the research experimental results and it's significant in the following manner:

a. The first step explained how the result of base learners impacted the construction of the stacked ensemble model. The model's selection was based on the SAID technique explained 3.6. The result led to two construction of stacked ensemble models for this research work.
b. Thereafter, the multi-class weather features performance was measured to determine the rate of their contribution to the overall experiment. It was seen that gradient feature had the lowest performance in both experiments. This implies that the feature contribution to the overall performance is the lowest while contrast features shows the reverse in experiment II.
c. The multi-class weather still images recognition performance was measured based on the two constructed stacked ensemble models. The results show over 90% recognition rate for the rainy images while the others i.e. cloudy, sunny and sunrise shows over 80% recognition rate in both experiments respectively.
d. Finally, we compared the result of the combine/meta-learner of the constructed stacked ensemble model with its base learners. The result shows that meta-learner outperformed its base learners in both experiments.

# CHAPTER FIVE
# CONCLUSION AND RECOMMENDATION

This chapter presents the summary of the work done to achieve the research goals/objectives, the limitations of the research and provide suggestion for future work.

## 5.1 CONCLUSION

The contribution of this research work was derived from the steps taken to answer the research question presented in Section 1.4 of Chapter One : *How to develop an intelligent computer vision system that can accurately recognize different weather condition that can replace sensor devices?* This work provide the following contribution:

1. **Dataset** : Given the absence of multi-class weather dataset that could meet experimental requirements i.e. being formed from unobstructed opaque cloud coverage (OCC) weather images. An unobstructed OCC multi-class weather dataset was manually collected online and annotated and made public for use.

2. **SAID Diversity Concept**: As established in Section 3.4, the diversity of any ensemble base learners is crucial to the successful of the meta-model. Since there was no established standard defined for diversity, This research developed a new concept for diversity called Selection Base on Intuition, Accuracy and Diversity (SAID). SAID uses a minimum 10% variance between base learners to determine how base learners will be combined. The concept result is promising.

3. **Stacked Ensemble Model:** Although the techniques of stacked ensemble principle has been around for decades, this research work presents the first application of this technique to weather recognition from still images.

   Furthermore, when the result of the meta-learner of the constructed stacked ensemble model is compared with the individual base model's result that makeup the stacked ensemble model. It was observed that the meta-learner outperformed its base-learners.

In conclusion, the research presents academic and practitioners a new insight into diversity of heterogeneous stacked ensemble methods for solving the challenges of weather recognition from still images.

## 5.2 RECOMMENDATION

This section highlighted the challenges faced during the course of this research work and also provide suggestions on how to tackle such challenges in the future.

### 5.2.1 LIMITATION OF THE STUDY

This research work is not void of some limitations. Without a doubt, a few strategies that might have helped in getting better execution have not been attempted. For instance, in the study there was no use of any data geometric transformation or segmentation techniques to our input. Toward the start of the investigation phase, an endeavor was made to apply data geometric transformation concept of extricating crops from the weather images; specifically 5 cropped images were mined and separated: the focal aspect of the image and the four(4) edges. In any case, during the training it was seen that this was not valueable and, moreover, it was observed to be causing overfitting issues. As a result, this process of training was immediately halted because of  the enormous computation time it would have required for successful training of the model. Another justification for halting the training process was concluded because of the presence of such a large number of cropped sky images from the same images whose features might not contributes to the recognition or identification of the entire images during the validation phase. In any case, no further attempts were made once the working architecture was identified.

Another impediment of this research work can be seen in the dataset. In reality, the dataset shows various deluding images that subscribes in bringing down the performance results; for example, vague labelled images, poor images quality, and dark images.

### 5.2.2 RECOMMENDATION FOR FUTURE WORK

In view of the result of this research work, we give suggestions for future work;

We recommend improving the quality of images that made up the dataset,while on the other hand, neglecting those images that failed to offer any valuable information to the task but lend itself to noise properties. As observed during the cross-validation stage, the quality of the images contributes a lot to the final outcome, particularly when training and validating the set that are not evenly stable in connection with the distribution of different scenarios. Therefore, the approach of either manual or programmed (automatic) recognition of these weather images can be actualized. For example, with respect to the manual recognition, human engagement approach could be utlilized. Furthermore, internet surfers across the globe could be asked to recognize climate conditions among a category of available weather images or, given a climatic condition and a catalogs of different weather images, this could be asked to identifyapproached by distinguish from the catalogs of different weather images the most representated of a particular climate to the least representated ones, in their view, the better representation of any given weather condition will presented. With the aid of programming, automation recognition approach utilized software driven programming language to eliminate dark and the non outdoor images or, through the techniques of OCR algorithms, to elimate images with text content.

The further recommends that research should be carried out to see the performance effect of increasing the number of classes to be classified for this task. For example, increasing the number of images per class in the dataset could lead to better model understanding of the class image. Hence, it is imagined that this could improve model performances because the method utilised iterative learning approach of understanding the class dataset, in the manner in which the class climatic structures are represented, as a result, this would made classfication of weather images task easy and prevent the conflicting presence of too different characteristics that exhibits itself in more than one weather conditions, as it happens with the no-rain class.

Based on the findings of this study, It would be fascinating to explore a different kind of input source, for example, a video input will change image state to dynamic state. In this manner, the dynamic of rain downpour could be better measured and likely more effectively detected.

In conclusion, it might be helpful, for gaining an additional insight by considering feature visualization to see which features that can used to segregate between different classes.

## REFERENCES


- E.M. Fischer and R.Knutti, “Anthropogenic contribution to global occurrence of heavy-precipitation and high-temperature extremes”, Nature Climate Change 5, 560 (2015).
- C. Lu, D. Lin, J. Jia, and C.-K. Tang, June 2014 “Two-class weather classification,” in Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’14), pp. 3718–3725, Columbus, Ohio, USA
- F. Nashashibi, R. de Charette, and A. Lia, 2010, “Detection of unfocused raindrops on a windscreen using low level image processing,” in Proceedings of the 11th International Conference on Control Automation Robotics & Vision (ICARCV ‘10), pp.1410-1415, Singapore, December 2010.
- H. Kurihata, T. Takahashi, I. Ide et al., 1963, “Rainy weather recognition from in-vehicle camera images for driver assistance,” in Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 205–210, IEEE,
- H. Katsura, J. Miura, M. Hild, and Y. Shirai, “A view-based outdoor navigation using object recognition robust to changes of weather and seasons,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’03), vol. 3, pp. 2974–2979
- H. Woo, Y. M. Jung, J.-G. Kim, and J. K. Seo, 2010, “Environmentally robust motion detection for video surveillance,” IEEE Transactions on Image Processing, vol. 19, no. 11, pp. 2838–2848.
- K. Derpanis, M. Lecce, K. Daniilidis, and R. Wildes. Dy-namic scene understanding: The role of orientation features in space and time in scene classification. InCVPR, 2012.

- J. Bossu, N. Hautière, and J.-P. Tarel, Rain or snow detection in image sequences through use of a histogram of orientation of streaks, International journal of computer vision 93, 348 (2011)
- J. Mairal, F. Bach, J. Ponce, Task-driven dictionary learning, IEEE Trans. PatternAnal. Mach. Intell. 34 (2012) 791–804.
- S. Gao, I.W. Tsang, L.-T. Chia, Sparse representation with kernels, IEEE Trans.Image Process. 22 (2013) 423–434.
- K. Garg and S. K. Nayar, when does a camera see rain? in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, Vol. 2 (IEEE, 2005) pp. 1067–1074.

- M. Elhoseiny, S. Huang, and A. Elgammal, Weather classification with deep convolutional neural networks, in Image Processing (ICIP), 2015 IEEE International Conference on (IEEE, 2015) pp. 3349–3353.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in Advances in neural information processing systems (2012) pp. 1097–1105.
- Z. Zhu, L. Zhuo, P. Qu, K. Zhou, and J. Zhang, Extreme weather recognition using convolutional neural networks, in Multimedia (ISM), 2016 IEEE International Symposium on (IEEE, 2016) pp. 621–625
- Zhi-Hua Zhou, Ensemble Methods: Foundation and Algorithms, Chapman & Hall/CRC Machine Learning & Pattern Recognition Series, International Standard Book Number-13: 978-1-4398-3005-5, 2012

- C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A.Rabinovich, Going deeper with convolutions, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015) pp. 1–9
- C. Zhang and Y. Ma (eds.), Ensemble Machine Learning: Methods and Applications, DOI 10.1007/978-1-4419-9326-7 1, © Springer Science+Business Media, LLC 2012
- T. G. Dietterich, "Ensemble methods in machine learning," in Multiple Classifier Systems. Springer, 2000, pp. 1–15
- R. Kohavi and D. H. Wolpert, "Bias plus variance decomposition for zero-one loss functions," in Proc. International Conference on Machine Learning (ICML'96), 1996, pp. 275–283.
- L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
- Y. Ren, P. N. Suganthan, and N. Srikanth, "Ensemble methods for wind and solar power forecasting: A state-of-the-art review," Renewable  Sustain. Energy Rev., vol. 50, pp. 82–91, Oct. 2015.
- A. Krogh, J. Vedelsby et al., "Neural network ensembles, cross validation, and active learning," Advances in Neural Information Processing Systems, pp. 231–238, 1995.
- G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: a survey and categorisation," Information Fusion, vol. 6, no. 1, pp. 5–20, 2005.
- Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," Neural Computation, vol. 4, no. 1, pp. 1–58, 1992.
- V. Pisetta, "New insights into decision tree ensembles," Ph.D. dissertation, Lyon 2. Learning, vol. 51, no. 2, pp. 115–135, 2003.

- L. Zhang, Y. Ren, and P. N. Suganthan, “Towards generating random forest with extremely randomized trees,” in Proc. IEEE International Joint Conference on Neural Networks (IJCNN’14), Beijing, China, Jul. 2014
- L. Zhang and P. N. Suganthan, “Oblique decision tree ensemble via multisurface proximal support vector machine,” IEEE Trans. Cybern., pp. 2168–2267, Nov. 2014.
- L. Breiman, “Bias, variance, and arcing classifiers,” University of California, Berkeley, CA, Tech. Rep. 460, 1996.
- P. Domingos, “A unified bias-variance decomposition,” in Proc. International Conference on Machine Learning (ICML’00), 2000, pp. 231–238.
- E. Kleinberg, “Stochastic discrimination,” Annals of Mathematics and Artificial intelligence, vol. 1, no. 1, pp. 207–239, 1990
- R. E. Schapire and Y. Freund, “Boosting the margin: A new explanation for the effectiveness of voting methods,” The Annals of Statistics, vol. 26, pp. 322–330, 1998.
- Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” in Proc. International Conference on Machine Learning (ICML.96), vol. 96, 1996, pp. 148–156.
- C.-X. Zhang and J.-S. Zhang, “RotBoost: A technique for combining rotation forest and adaboost,” Pattern Recognition Letters, vol. 29, no. 10, pp. 1524–1536, 2008
- L. Breiman, “Bagging predictors,” Machine Learning, vol. 24, no. 2, pp. 123–140, 1996.
- T. K. Ho, “The random subspace method for constructing decision forests,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 8, pp. 832–844, 1998.
- C. Tan, M. Li, and X. Qin, “Random subspace regression ensemble for near-infrared spectroscopic calibration of tobacco samples,” Analytical Sciences, vol. 24, no. 5, pp. 647–654, 2008.

- P. Viola and M. Jones. Rapid object detection using a boosted cascade ofsimple features. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 511–518, Kauai, HI,2001.
- P. Viola and M. Jones. Robust real-time object detection. International Journal of Computer Vision, 57(2):137–154, 2004.
- F.-J. Huang, Z.-H. Zhou, H.-J. Zhang, and T. Chen. Pose invariant face recognition. In Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, pages 245–250, Grenoble, France,2000.
- S. Avidan. Ensemble tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(2):261–271, 2007.
- I. Corona, G. Giacinto, C. Mazzariello, F. Roli, and C. Sansone. Informationfusion for computer security: State of the art and open issues.Informa-tion Fusion, 10(4):274–284, 2009
- G. Giacinto, F. Roli, and L. Didaci. Fusion of multiple classifiers for intru-sion detection in computer networks.Pattern Recognition Letters, 24(12):1795–1803, 2003.
- G. Giacinto, R. Perdisci, M. D. Rio, and F. Roli. Intrusion detection in com-puter networks by a modular ensemble of one-class classifiers.Informa-tion Fusion, 9(1):69–82, 2008
- Z.-H. Zhou, Y. Jiang, Y.-B. Yang, and S.-F. Chen. Lung cancer cell identifica-tion based on artificial neural network ensembles.Artificial Intelligencein Medicine, 24(1):25–36, 2002a
- R. Polikar, A.Topalis,D.Parikh,D.Green,J.Frymiare,J.Kounios,andC.M.Clark. An ensemble-based data fusion approach for early diagnosis of Alzheimer’s disease. Information Fusion, 9(1):83–95, 2008

- P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo. Distributed data min-ing in credit card fraud detection.IEEE Intelligent Systems, 14(6):67–74,1999.
- S. Panigrahi, A. Kundu, S. Sural, and A. K. Majumdar. Credit card fraud detection: A fusion approach using Dempster-Shafer theory and Bayesianlearning.Information Fusion, 10(4):354–363, 2009
- K. Goebel, M. Krok, and H. Sutherland. Diagnostic information fusion: Re-quirements flowdown and interface issues. InProceedings of the IEEEAerospace Conference, volume 6, pages 155–162, Big Sky, MT, 2000
- W. Yan and F. Xue. Jet engine gas path fault diagnosis using dynamic fusionof multiple classifiers. In Proceedings of the International Joint Conferenceon Neural Networks, pages 1585–1591, Hong Kong, China, 2008.
- D. West, S. Dellana, and J. Qian. Neural network ensemble strategies for fi-nancial decision applications. Computers & Operations Research, 32(10):2543–2559, 2005.
- M. B. Ara´ujo and M. New. Ensemble forecasting of species distributions. Trends in Ecology & Evolution, 22(1):42–47, 2007.
- J. W. Taylor and R. Buizza. Neural network load forecasting with weather ensemble predictions. IEEE Transactions on Power Systems, 17(3):626–632, 2002.
- J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. K´egl. Aggregate features and AdaBoost for music classification. Machine Learning, 65(2-3):473–484, 2006.
- I.Maqsood,M.R.Khan,andA.Abraham.Anensembleofneuralnetworksfor weather forecasting.Neural Computing & Applications, 13(2):112–122, 2004.
- T. Gneiting and A. E. Raftery. Atmospheric science: Weather forecasting with ensemble methods. Science, 310(5746):248–249, 2005.

- A. C. Tan, D. Gilbert, and Y. Deville. Multi-class protein fold classification using a new ensemble machine learning approach. Genome Informatics,14:206–217, 2003.
- H. B. Shen and K. C. Chou. Ensemble classifier for protein fold pattern recognition. Bioinformatics, 22(14):1717–1722, 2006
- M. Kearns and L. G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. In Proceedings of the 21st Annual ACM Symposium on Theory of Computing, pages 433–444, Seattle, WA, 1989.
- Kirchner, M., Timm, W., Fong, P., Wangemann, P., Steen, H.: Non-linear classification foron-the-fly fractional mass filtering and targeted precursor fragmentation in mass spectrometryexperiments. Bioinformatics26(6), 791 (2010)
- Riddick, G., Song, H., Ahn, S., Walling, J., Borges-Rivera, D., Zhang, W., Fine, H.: Predictingin vitro drug sensitivity using random forests. Bioinformatics27(2), 220 (2011)
- B. Efron and R. Tibshirani.An Introduction to the Bootstrap.Chapman&Hall, New York, NY, 1993
- Buntine, W., A Theory of Learning Classification Rules. Doctoral dissertation.School of Computing Science, University of Technology. Sydney. Australia,1990
- Bauer, E. and Kohavi, R., “An Empirical Comparison of Voting ClassificationAlgorithms: Bagging, Boosting, and Variants”. Machine Learning, 35: 1-38,1999.
- Opitz, D. and Maclin, R., Popular Ensemble Methods: An Empirical Study,Journal of Artificial Research, 11: 169-198, 1999.
- Wolpert, D.H., Stacked Generalization, Neural Networks, Vol. 5, pp. 241-259, Pergamon Press, 1992

- P. Smyth and D. Wolpert. Stacked density estimation. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, Advances in Neural Information Processing Systems 10, pages 668–674. MIT Press, Cambridge, MA, 1998.
- K. M. Ting and I. H. Witten. Issues in stacked generalization. Journal of Artificial Intelligence Research, 10:271–289, 1999.
- A. K. Seewald. How to make stacking better and faster while also taking care of an unknown weakness. In Proceedings of the 19th International Conference on Machine Learning, pages 554–561, Sydney, Australia, 2002.
- B. Clarke. Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. Journal of Machine Learning Re-search, 4:683–712, 2003.

**Appendix A:** Sample python source codes

**A1:** Listing One- Image Downloading Script

```
"""
Filename: image_download.py

Author: Gbeminiyi Ajayi

Description: This script is generic for all image downloads across different platform. This requires python api module to work. The script below imports flickr python api developed by James Clarke used as interface binder to connect to flickr account.
"""
#import python modules

import  flickr
import  urllib, urlparse
import  os, sys

# obtain the commandline augment
if len(sys.argv)>1:
    tag = sys.argv[1]
else:
    print 'no tag specified'

# downloading image data
f = flickr.photos_search(tags=tag)
urllist = [] #store a list of what was downloaded

# downloading images
for k in f:
    url = k.getURL(size='Medium', urlType='source')
    urllist.append(url)
    image = urllib.URLopener()
    image.retrieve(url, os.path.basename(urlparse.urlparse(url).path))
    print 'downloading:', url
```

**A2:** Listing: Image Resizing Script

```
"""
Filename: imresize.py

Author: Gbeminiyi Ajayi

Description: This script is part of the pre-processing stage of weather images by resizing different images into uniform image dimension of 300px x 240px.
"""
from PIL import Image
import os,sys

def imageResize(path,prefix='IMG'):
    """Resize the image to 300 x 240 jpg and store them in a new directory"""

    print "[*] Fetching jpg image from "+path
    imList = [os.path.join(path,f) for f in os.listdir(path) if f.endswith('.jpg')]
    i = 0
    print "[+] Saving Images......."
    for image in imList:
        Image.open(image).resize((300,240)).save(image)
        print "..."+image +" saved resized "
        i+=1



if __name__ == '__main__':
    print '********************************'
    print 'Starting Image Resizing Script'
    print '********************************'
    imageResize(sys.argv[1])
```

## **A3:** Listing Three: Feature Extraction Script

```
"""
Filename: wfeature.py

Author: Gbeminiyi Ajayi

Description: Pipeline functions for extracting different weather features from each image. The features are stored as object using python default library called pickle
"""

# initialising of python modules
from PIL import Image, ImageEnhance
from pylab import *
from skimage import color, feature
from scipy.ndimage import filters
import sift, colorsys


# function to extract HSV
def extract_hsv_features(imList):
        hfeatures = sfeatures = vfeatures = zeros([len(imList), 128])
        # assigning numerica label 0 - H, 1- S, V-2
        h_labels = slabels = vlabels = zeros([len(imList), ])
        for i, im in enumerate(imList):
            print "...processing ", im
            im = color.rgb2hsv(array(Image.open(im)))
            h, s, v = im[:,:,0], im[:,:,1], im[:,:,2]
            H_hist, h1 = histogram(h.flatten(), bins=128,
     density=True)
            hfeatures[i] = H_hist
            h_labels[i] = 0
            S_hist, h2 = histogram(s.flatten(), bins=128,
     density=True)
            sfeatures[i] = S_hist
            slabels[i] = 1
```

```
            V_hist, h3 = histogram(v.flatten(), bins=128,
    density=True)
            vfeatures[i] = V_hist
            vlabels[i] = 2
            print " finished"

        return hfeatures, sfeatures, vfeatures, h_labels, slabels,
    vlabels

#function to extract  gradient functon based Sobex derivative
def extract_gradient_features(imList):
        # assign label 4 for gradient features
        labels = zeros([len(imList), ])
        features = zeros([len(imList), 128])
        for i, im in enumerate(imList):
            im = array(Image.open(im).convert("L"))
            # print im.shape
            # Sobel derivative filters
            imx = zeros(im.shape)
            filters.sobel(im, 1, imx)

            imy = zeros(im.shape)
            filters.sobel(im, 0, imy)

            magnitude = sqrt(imx ** 2 + imy ** 2)
            hist, _ = histogram(magnitude.flatten(), bins=128,
    density=True)
            features[i] = hist
            labels[i] = 3

        return features, labels

# function to extract SIFT features
def extract_sift_features(imList):
        # assign label 5 for each feature
```

```
        descriptor = []

        features = zeros([len(imList), 128])
        labels = zeros([len(imList), ])
        for i, feat in enumerate(imList):
            sift.process_image(feat, feat[:-3]+'sift')
            loc1, descr = sift.read_features_from_file(feat[:-
    3]+'sift')
            # hist, _ = histogram(descr.flatten(), bins=128,
    density=True)
            # print descr.shape
            descriptor.append(descr.flatten())
            # features[i] = hist
            labels[i] = 4

        return array(descriptor), labels


# function to extract SIFT features
def extract_contrast_features(imList):
        # assign label 6 for each features
        labels = zeros([len(imList), ])
        features = zeros([len(imList), 128])
        for i, img in enumerate(imList):
            im = Image.open(img)
            img = array(im)
            # obtained max and min intensity
            max = img.max() * 1.0
            min = img.min() * 1.0

            # compute the image contrast factor
            factor = (max - min) / (max + min)

            enhancer = ImageEnhance.Contrast(im)
            im2 = enhancer.enhance(factor)
```

```
            hist, _ = histogram(im2, bins=128,  density=True)
            features[i] = hist
            labels[i] = 5

        return features, labels



#function to extract LBP
def extract_lbp_features(imList, numPoint=24, radius=8, eps=1e7):
        """ Extraction of local binary pattern with radius 8 x 8
    boxes division"""
        features = zeros([len(imList), 128])
        labels = zeros([len(imList), ]) # assign 7
        for i, im in enumerate(imList):
            image = array(Image.open(im).convert("L"))
            lbp = feature.local_binary_pattern(image, numPoint,
    radius, method='uniform')
            hist, _ = histogram(lbp.flatten(),bins=128, density=True)
            features[i] =hist
            labels[i] = 6
            # # normalise the histogram
            # hist = hist.astype('float')
            # hist /= (hist.sum() + eps)

        return features, labels
```

**A4. Listing Four: Feature Object Extractor**

```
"""
Filename: extract.py

Author: Gbeminiyi Ajayi

Description: This script reference wfeature.py to extract all the weather features from reference dataset. The extracted features are saved as pickle object
"""

#import python modules
import os, pickle
from pylab import *
import wfeatures as ft

#specify path to the dataset
path = "dataset2"

#create image list and label from the images
imlist =[os.path.join(path, f) for f in os.listdir(path) if
f.endswith('.jpg')]
labels = [im.split('/')[-1][:2] for im in imlist]


print "Processing HSV features....."
h, s, v, _, _, _ = ft.extract_hsv_features(imlist)

print "Processing gradient features....."
grad, _ = ft.extract_gradient_features(imlist)

print "Processing SIFT features....."
sf, _ = ft.extract_sift_features(imlist)

print "Processing Contrast features....."
contrast, _= ft.extract_contrast_features(imlist)
```

```
#
print "Processing LBP features....."
lbp, _ = ft.extract_lbp_features(imlist)

#Save weather features as pickle object
with open('features.pkl','wb') as f:
    pickle.dump(h, f)
    pickle.dump(s, f)
    pickle.dump(v, f)
    pickle.dump(grad, f)
    pickle.dump(sf, f)
    pickle.dump(contrast, f)
    pickle.dump(lbp, f)
    pickle.dump(labels, f)


print "Features extracted and saved as features.pkl"
```

## A5. Listing Five: Stacked Model Script

```
"""
Filename: stack.py

Author: Gbeminiyi Ajayi

Description: The script implements stacked ensemble method described in chapter three. This script imports all the base learners and the meta-classifier. The extracted features stored as object are referenced by importing into the memory for creating trained library for classification of weather images. The dataset was divided into dev and test. The dev dataset was training the heterogenous stack algorithm while the test is the unseen dataset. Finally, the script output result is as follows:
    1. Feature Performance, and
    2. Weather image classification.
"""
```

```
#import sklearn libraries and other custom modules
from sklearn.cross_validation import StratifiedKFold
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
from sklearn.model_selection import cross_val_score, train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
import xgboost as xgb
from sklearn.ensemble import RandomForestClassifier,
ExtraTreesClassifier, GradientBoostingClassifier, AdaBoostClassifier
import matplotlib.pyplot as plt
import numpy as np
import pickle
import itertools


# creation of stacked classifier
def stack_classifier(data, label):

    X, y = data, label

    # The DEV SET will be used for all training and validation
purposes. 80% of the images were used .
    # The TEST SET will never be used for training, it is the unseen
test
    dev_cut = len(y) * 4/5
    X_dev = X[:dev_cut]
    Y_dev = y[:dev_cut]
```

```
    X_test = X[dev_cut:]
    Y_test = y[dev_cut:]

    n_trees = 100
    n_folds = 5


    # Ready for cross validation
    skf = list(StratifiedKFold(Y_dev, n_folds))


    # Our level 0 classifier i.e. Base learners
    clfs = [

        RandomForestClassifier(n_estimators=n_trees, criterion='gini',
n_jobs=-1),
        # ExtraTreesClassifier(n_estimators=n_trees, criterion='gini',
n_jobs=-1),
        KNeighborsClassifier(n_neighbors=5),
        SVC(kernel='rbf'),
        # GaussianNB(),
        # GradientBoostingClassifier(n_estimators=n_trees)

    ]

    # Pre-allocate the data
    stack_train = np.zeros((X_dev.shape[0], len(clfs)))
    stack_test = np.zeros((X_test.shape[0], len(clfs)))


    # For each classifier, we train the number of fold times
(=len(skf))

    for j, clf in enumerate(clfs):
        print "Training classifier [%s]" %(j)
```

```
        stack_test_j = np.zeros((X_test.shape[0], len(skf)))
        for i, (train_index, cv_index) in enumerate(skf):
            print "Fold [%s]" %(i)

            # This is the training and validation set
            X_train = X_dev[train_index]
            Y_train = Y_dev[train_index]

            X_cv = X_dev[cv_index]
            Y_cv = Y_dev[cv_index]

            clf.fit(X_train, Y_train)

            # This output will be the basis for our blended classifier
to train against
            # which is also the output of our classifier
            stack_train[cv_index, j] = clf.predict(X_cv)
            stack_test_j[:, i] = clf.predict(X_test)

        # Get the mean predictions of the cross validation sets
        stack_test[:, j] = stack_test_j.mean(1)

    print 'Y_dev.shape = %s' %(Y_dev.shape)

    return stack_train, stack_test, Y_dev, Y_test


def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
```

```
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]),
range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('Labels')
    plt.xlabel('Predictions')



def classifier (features, Y_dev, test, Y_test,
weight=False,cnf_matrix=False, feature_name=None):
    if cnf_matrix is True:
        clf = GradientBoostingClassifier(n_estimators=100)
        # clf = SVC(kernel='rbf')
```

```
else:
    clf = SVC(kernel='linear')



# train the classifier
clf.fit(features, Y_dev)

# xg_train = xgb.DMatrix(features, label=Y_dev)
# xg_test = xgb.DMatrix(test, label=Y_test)

# # setup parameter for the xgboost
# param = {}
#
# param['eta'] = 0.1
# param['max_depth'] = 6
# param['silent'] = 1
# param['nthread'] = 4
# param['num_class'] = 6
#
# watchlist = [(xg_train, 'train'), (xg_test, 'test')]
# num_round = 5

# clf = xgb.train(param,xg_train,num_round, watchlist)



# Predict now
Y_predict = clf.predict(test)
score = metrics.accuracy_score(Y_test, Y_predict)
if weight is True:
    print "%s Weight = %s " % (feature_name, clf.coef_)
    print "%s Accurary = %s" % (feature_name, score)

# Features Ranking
if cnf_matrix is True:
    # Compute confusion matrix
```

```
        cnf_matrix = metrics.confusion_matrix(Y_test, Y_predict)
        np.set_printoptions(precision=2)

        # Plot normalized confusion matrix
        plt.figure()
        plot_confusion_matrix(cnf_matrix, classes=class_names,
normalize=True,
                              title='Confusion Matrix')
        # plot_features(features)

    else:
        return score

def plot_features(scores):
    label = ('Hue', 'Saturation', 'Value', 'Gradient', 'LBP',
'Contrast')
    y_pos = np.arange(len(label))
    plt.figure()
    plt.title('Feature Accuracy')
    plt.bar(y_pos, scores, color='b', align='center', alpha=0.5)
    plt.xticks(y_pos, label)
    plt.ylabel(' Classification Accuracy')


def other_method(X, y):
    print "training lenght ", len(X)
    print "label length ", len(y)

    n_trees = 100
    clfs = [

        KNeighborsClassifier(n_neighbors=5),
        SVC(kernel='rbf'),
        GaussianNB(),
```

```
        RandomForestClassifier(n_estimators=n_trees, criterion='gini',
n_jobs=-1),
        ExtraTreesClassifier(n_estimators=n_trees * 2,
criterion='gini', n_jobs=-1),
        # AdaBoostClassifier(n_estimators=n_trees)
        GradientBoostingClassifier(n_estimators=n_trees)
    ]


    for clf, clf_label in zip(clfs,['KNN','SVM','Native Baye','Random
Forest',' Extra Tree', 'Gradient Boost']):
        # clf.fit(X_train, y_train)
        # score = clf.score(X_test,y_test)
        score = cross_val_score(clf, X=X, y=y, cv=5,
scoring='accuracy')
        print "%s Accuracy: %0.2f (%0.2f)" % (clf_label, score.mean(),
score.std())



if __name__ == '__main__':

    # extract features from the saved pickle data
    with open('features2.pkl', 'rb') as f:
        hfeatures = pickle.load(f)
        sfeatures = pickle.load(f)
        vfeatures = pickle.load(f)
        gfeatures = pickle.load(f)
        # sifeatures = pickle.load(f)
        cfeatures = pickle.load(f)
        lfeatures = pickle.load(f)
        labels = pickle.load(f)

    # convert label string to numeric
    label_encoder = LabelEncoder()
```

```
    label_encoder.fit(labels)
    labels = label_encoder.transform(labels)

    original_features =
np.concatenate((hfeatures,sfeatures,vfeatures,gfeatures,cfeatures,lfea
tures), axis=1)

    class_names = label_encoder.classes_

    # Train each features
    hue_stack_train, hue_stack_test, Y_dev, Y_test =
stack_classifier(hfeatures, labels)
    sat_stack_train, sat_stack_test, Y_dev, Y_test =
stack_classifier(sfeatures, labels)
    val_stack_train, val_stack_test, Y_dev, Y_test =
stack_classifier(vfeatures, labels)
    gra_stack_train, gra_stack_test, Y_dev, Y_test =
stack_classifier(gfeatures, labels)
    # sif_stack_train, sif_stack_test, Y_dev, Y_test =
stack_classifier(sifeatures, labels)
    con_stack_train, con_stack_test, Y_dev, Y_test =
stack_classifier(cfeatures, labels)
    lbp_stack_train, lbp_stack_test, Y_dev, Y_test =
stack_classifier(lfeatures, labels)

    # generate individual feature accuracy
    hue = classifier(hue_stack_train, Y_dev, hue_stack_test,  Y_test,
feature_name='Hue')
    sat = classifier(sat_stack_train, Y_dev, sat_stack_test,  Y_test,
feature_name='Saturation')
    va = classifier(val_stack_train, Y_dev, val_stack_test,  Y_test,
feature_name='Value')
    gra = classifier(gra_stack_train, Y_dev, gra_stack_test,  Y_test,
feature_name='Gradient')
```

```
    # classifier(sif_stack_train, Y_dev, sif_stack_test,  Y_test,
feature_name='SIFT')
    lbp = classifier(lbp_stack_train, Y_dev, lbp_stack_test,  Y_test,
feature_name='LBP')
    con = classifier(con_stack_train, Y_dev, con_stack_test,  Y_test,
feature_name='Contrast')

    # concatenate the stacked trained features and test features
    features = np.concatenate((hue_stack_train, sat_stack_train,
val_stack_train, gra_stack_train,  con_stack_train, lbp_stack_train),
axis=1)

    test = np.concatenate((hue_stack_test, sat_stack_test,
val_stack_test, gra_stack_test, con_stack_test, lbp_stack_test),
axis=1)


    # start blending
    classifier(features, Y_dev, test, Y_test, cnf_matrix=True,
feature_name="Stacking")

    other_method(original_features, labels)


    plot_features([hue, sat, va, gra, lbp, con])

    plt.show()
    # # Other methods
    # # print len(labels)
    # print len(features)
```

UNISA university of south africa

## UNISA SOE ETHICS REVIEW COMMITTEE

Date:13/12/2018

Dear Mr Gbeminiyi Oluwafemi Ajayi

| |
|---|
| **Decision: Ethics Approval from 13/12/2018 to 13/12/2021** |

| |
|---|
| ERC Reference # : 2018/CSET_SOE/GOA/001 |
| Name : Mr Gbeminiyi Oluwafemi Ajayi |
| Student #: 50374443 |
| Staff #: N/A |

**Researcher(s):** Name: Mr Gbeminiyi Oluwafemi Ajayi
Address:
E-mail address: 50374443@mylife.unisa.ac.za,
telephone #: +2348038478749

**Supervisor (s)**: Name: Zenghui Wang
Email: wangz@unisa.ac.za; Telephone: 011 471 3513

| |
|---|
| **Working title of research:** |
| **Weather classification from still images using ensemble method.** |

**Qualification:** Masters

Thank you for the application for research ethics clearance by the Unisa SOE Ethics Review Committee for the above mentioned research. Ethics approval is granted for 3 years.

*The **low risk application** was **reviewed** by the SOE Ethics Review Committee on 13/12/2018 in compliance with the Unisa Policy on Research Ethics and the Standard Operating Procedure on Research Ethics Risk Assessment. The decision was approved on 13/12/2018.*

The proposed research may now commence with the provisions that:

1. The researcher(s) will ensure that the research project adheres to the values and principles expressed in the UNISA Policy on Research Ethics.
2. Any adverse circumstance arising in the undertaking of the research project that is relevant to the ethicality of the study should be communicated in writing to the SOE Committee.
3. The researcher(s) will conduct the study according to the methods and procedures set out in the approved application.
4. Any changes that can affect the study-related risks for the research participants, particularly in terms of assurances made with regards to the protection of participants' privacy and the confidentiality of the data, should be reported to the Committee in writing, accompanied by a progress report.
5. The researcher will ensure that the research project adheres to any applicable national legislation, professional codes of conduct, institutional guidelines and scientific standards relevant to the specific field of study. Adherence to the following South African legislation is important, if applicable: Protection of Personal Information Act, no 4 of 2013; Children's act no 38 of 2005 and the National Health Act, no 61 of 2003.
6. Only de-identified research data may be used for secondary research purposes in future on condition that the research objectives are similar to those of the original research. Secondary use of identifiable human research data require additional ethics clearance.
7. No field work activities may continue after the expiry date 13/12/2021. Submission of a completed research ethics progress report will constitute an application for renewal of Ethics Research Committee approval.
8. Field work activities may only commence from the date on this ethics certificate.
9. [Permission to conduct research involving UNISA employees, students and data should be obtained from the Research Permissions Subcommittee (RPSC) prior to commencing field work.] AND/OR
10. [Permission to conduct this research should be obtained from the [company, CE organisation, DoE, etc name] prior to commencing field work.]



University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za

Add any other conditions if relevant.

*Note:*

*The reference number* ***2018/CSET_SOE/GOA/001*** *should be clearly indicated on all forms of communication with the intended research participants, as well as with the Committee.*

Yours sincerely,

Signature..........

Dr T Sithebe

Chair of SOE ERC

**E-mail: sithet@unisa.ac.za**

**Tel: (011) 429-3864**

Signature..........

Prof BB Mamba

Executive Dean : CSET

**E-mail: mambabb@unisa.ac.za**

**Tel: (011) 670-9230**

13/12/2018

F. Nemavhsro.

RECEIVED

2018 -12- 13



University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za

# *BBM English Language Scientific Editing Services*

I, **BELLITA BANDA** hereby confirm that I have proof read and edited the **MAGISTER TECHNOLOGIAE**

Titled

**Weather classification from still images using ensemble method**

**By**

**AJAYI GBEMINIYI OLUWAFEMI**

The Windows 'Tracking' System was used to reflect my comments and suggested corrections are given for the author to action.

During the process of the proof reading and editing, the following changes were recommended: punctuation, grammatical and sentence construction and how to improve on coherence of the document. In addition consistency in use of abbreviations, uniformity referencing style (intext and reference list) were given by the editor. Although greatest care was taken in editing this document, the final responsibility for the product rests with the author.

25/03/2020

Editor's signature Date

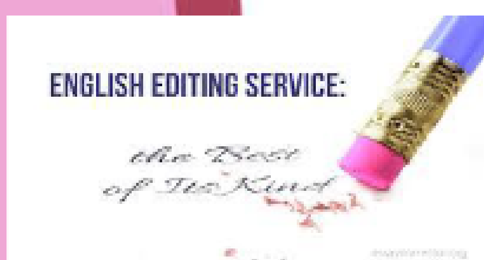
**Dr. B. Banda**

***Language and Writing and Editing Consultant***

***Intermediate Certificate in Business English & Communication (Pitman Qualification, UK)***

***Further Education in Teacher's Certificate (FETC, Zimbabwe)***

***Bachelor of Arts in English and Communication (Zimbabwe)***




**65 Smith Road, Alice, 5700, Eastern Cape South Africa**

bbanda@gmail.com
**Cell Number: 073 105 4838**