Apply CNN-XGBoost into Weather Image Recognition

Tran Quy Nam and Phi Cong Huy

Posts and Telecommunications Institute of Technology, Hanoi, Vietnam

Abstract. This study implements some hybrid deep learning network models for weather image classification. This study proposes to apply a hybrid model, namely CNN-XGBoost model to test its performance, in comparison with other simple Convolutional Neural Network (CNN) model with softmax and in addition with the other hybrid models, namely CNN-SVC, CNN-Decision Tree, CNN-AdaBoost, Multi-layer Perceptron Classifier which are all applied into the same problem of weather image classification. The models apply an identical test dataset which is a set of 11 different image classes that are collected from different resources of weather images with various kinds of weather phenomena. The test results show that the CNN-XGBoost gives the best results, which is suitable for application in evaluating weather images. The aim of this study is to check whether what kind of hybrid deep learning has the best performance in the problem of weather image classification, not focus on accuracy improvement of the deep learning models in classification problem.

Keywords: weather, image, CNN, XGBoost.

1 Introduction

The weather is changing nowadays which has a large impact on human life and socio-economic development of many countries in the world. The correct recognition of weather phenomenon is one of important factors to support our lives and nature development. There are some ways to recognize weather phenomenon, such as measurement of temperature, atmosphere, observational data collected by Doppler radar, weather satellites, and other instruments such as weather balloon to measure atmospheric parameters... The weather models use some mathematical and statistical equations, along with new and past weather data, to provide informative guidance. In computer science, the development of computer vision system has achieved great success in many areas, such as image processing with high accuracy has already got many applications in surveillance, navigation, driver assistance system…

The automatic methodology of weather image classification through AI (Artificial Intelligence) technology can help people to achieve sustainable development. The accurate processing and identification of weather images taken from drone or camera observation stations is an important method in weather forecasting, environmental assessments, warning dangerous transportation... In terms of environmental assessments, it is important to classify the respective weather phenomenon to alarm people before going outside, which help to give good guidance to people for wearing, traveling in the appropriate weathers. In addition, the high accurate recognition of weather image can help people to avoid negative effects or damages of natural disasters.

For weather forecasting, the correct recognition of image of weather phenomena that occurred the day before will also affect weather conditions for the next few days, thus lead to the exact assessment of environmental quality on weather. Furthermore, the high accuracy classifications of different weather phenomena have positive impacts on agriculture. The accurate recognition of weather phenomena can improve agricultural production. In transportation, the trustful assessment of weather phenomena has much influence on moving, transportation, and vehicle assistant driving systems. Therefore, it is becoming a significant issue to soon recognize the weather image in our daily lives. The automatic methodology of weather image classification through AI technology can help people to achieve sustainable development. The accurate processing and identification of weather images taken from drone or camera observation stations is an important method in social and economic development and our lives in the real world.

In this paper, we implement the combined models of CNN neural network with XGBoost algorithm to apply into a weather image phenomenon classification, which is a deep learning algorithm using a data set containing 6,862 images with 11 types of weather phenomena. Due to limited dataset of images and computing resources, this study does not aim to find the better and higher rate of accuracy of the deep learning models in classification problem. Instead, this study aim to prove that the proposed hybrid model, namely CNN-XGBoost can provide higher performance in term of accuracy measurement in comparisons with other models, such as simple CNN with softmax probability, and other hybrid models, namely CNN-SVC, CNN-Decision Tree, CNN-AdaBoost, CNN-Multi-layer Perceptron Classifier. The following paragraphs will describe the related works, methodology, experimental results and conclusion.

2 Related works

In fact, there have been many studies using machine learning models, deep learning models to identify weather images. In their papers, Xiao et al. [1] implemented a novel deep CNN that was named as MeteCNN for weather phenomena classification to provide good results. Their MeteCNN used VGG16 as the framework to build the proposed MeteCNN model. In fact, the MeteCNN discarded the fully connected layers (FCL) and then it added a global average pooling layer instead of max pooling layer before the softmax layer for classification task. Mohammad & Selvia [2] studied to classify weather images using CNN with Transfer Learning with four CNN architectures, namely MobileNetV2, VGG16, DenseNet201, and Xception to perform weather image classification. They used the transfer learning aimed to speed up the process of training models to get better and faster performance. They applied those 4 CNN architectures into the weather image which consists of six classes, namely cloudy, rainy, shine, sunrise, snowy, and foggy in dataset. The experiment result with 5-cross validation and 50 epochs showed that the Xception has the best average accuracy of 90.21% with 10,962 seconds of average training time and MobileNetV2 has the fastest average training time of 2,438 seconds with 83.51% of average accuracy. Mohamed et al. [3] introduced a novel framework to automatically extract the information from street-level images relying on deep learning and computer vision using a unified method without any pre-defined constraints in the processed images. They designed a pipeline of four deep convolutional neural network models, so-called WeatherNet, was trained, relying on residual learning using ResNet50 architecture, to extract various weather and visual conditions such as dawn/dusk, day and night for time detection, glare for lighting conditions, and clear, rainy, snowy, and foggy for weather conditions. Their WeatherNet showed strong performance in extracting this information from user-defined images or video streams. Khan et al. [4] studied some detection models which were focused on three weather conditions, namely clear, light snow, and heavy snow, as well as three surface conditions such as dry, snowy, wet/slushy. They applied them into several pre-trained CNN models, including AlexNet, GoogLeNet, and ResNet18 with proper modification via transfer learning. The best performance was achieved using ResNet18 architecture with an unprecedented overall detection accuracy of 97% for weather detection. Minhas et al. [5] studied weather prediction from real-world images via targeting classification using neural networks. In their article, the capabilities of a custom built driver simulator were explored specifically to simulate a wide range of weather conditions. The results indicated that the use of synthetic datasets in conjunction with real-world datasets could increase the training efficiency of the CNNs by as much as 74%.

In other researches on image recognition by CNN combined with other classifier algorithm, Thongsuwan S. et al [6] proposed a new deep learning model, namely Convolutional eXtreme Gradient Boosting (ConvXGB) for classification problems based on convolutional neural network and XGBoost algorithm. They designed their ConvXGB consists of several convolutional layers to learn the features of the input and, followed by XGBoost in the last layer for predicting the class labels. In the testing process, their ConvXGB model was applied into a dataset which were collected from the University of California at Irvine (UCI) Repository of machine learning. They concluded that the results of experiments on several datasets showed that the ConvXGB got slightly better results than CNN and XGBoost alone. In year of 2019, Thiyagarajan, S. [7] investigated the image processing in crack detection in construction engineering. The author used two-hybrid machine learning models and classified the concrete digital images. The aim of their research was to classify into cracks or non-cracks classes of images in concrete digital images. The Convolutional Neural Network was used to extract features from concrete pictures. And then, they used these extracted features as inputs for other machine learning models, namely Support Vector Machines (SVMs) and Extreme Gradient Boosting (XGBoost). The proposed method was evaluated on a collection of 40,000 real concrete images, and the experimental results showed that application of XGBoost classifier to CNN extracted image features included an advantage over SVM approach in the measurement of the accuracy score. Huang et al. [8] resolved the problem of low performance on different datasets on image processing classification, and also aim to resolve the real situation that there were very few specific large-scale datasets for training stages on image classification. They proposed a new combination classification model based on three pre-trained CNN models (VGG19, DenseNet169, and NASNetLarge) for processing the ImageNet database. The aim was to get better performance and tried to achieve higher classification accuracy. In their proposed model, the transfer learning model was based on each pre-trained model which was constructed as a possible classifier. In the next step, they figured out the best output of three possible classifiers which was concluded as the final classification choice. The implementation based on two waste image datasets on the same architectures of the proposed model had achieved the accuracy score at 96.5% and 94% relatively for classification problems. It means that their proposed model outperformed several other methods on image classification problems.

3 Methodology

In this study, we try to implement CNN-XGBoost architecture as the proposed network to apply for problem of weather image classification. For the combined model CNN-XGBoost, this study tests the model using the convolutional neural network for feature extraction and the XGBoost algorithm for image classification which are applied to the classification of faster images (see Fig. 1).

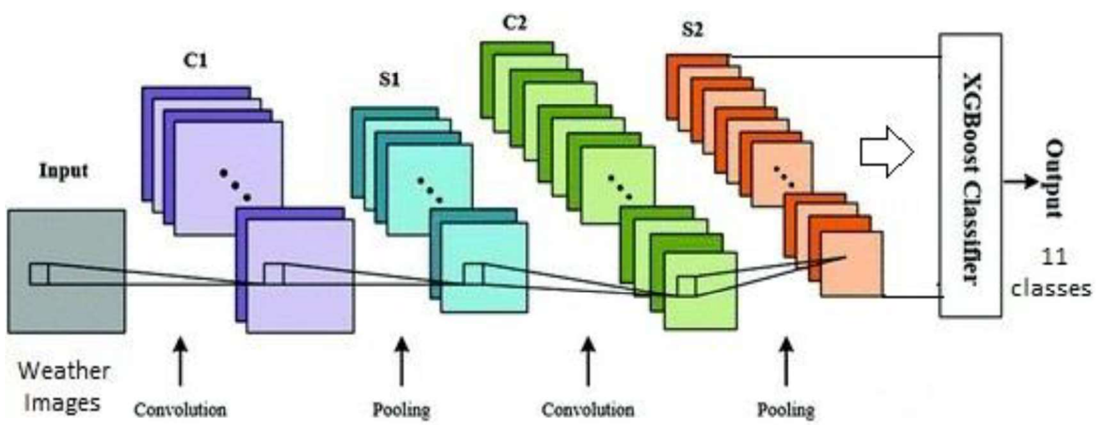

Fig. 1. XGBoost combined CNN network architecture

At the first stage for the feature extractor, we employ traditional Convolutional Neural Network model as the primary stage to extract features from the weather images. This CNN architecture of feature extractor has 3 Convolutional layers, the first with 128 filters, kernel size equal 3 and the other two layers with 64 nodes, each layer is followed by a Max Pooling with pooling size equal 2 and Dropout layers. The following layers are a Flatten layer and a Fully Connected layer with 128 nodes. A convolutional layer acquires a feature series by calculating the production between the receptive field and kernel. In this network, an activation function, namely Rectified Linear Unit (ReLU) function is added behind each convolutional layer. The

ISBN: 978-604-80-8083-9

optimizer is used as Adam, and a loss measure is categorical cross-entropy for multiple weather image classification.

A part from feature extractor, we also use this CNN architecture to classify the weather images. This part uses the ReLU (Rectified Linear Unit) for all the layers, except for the output layer where the softmax function was used to classify the weather images [see Fig. 2].

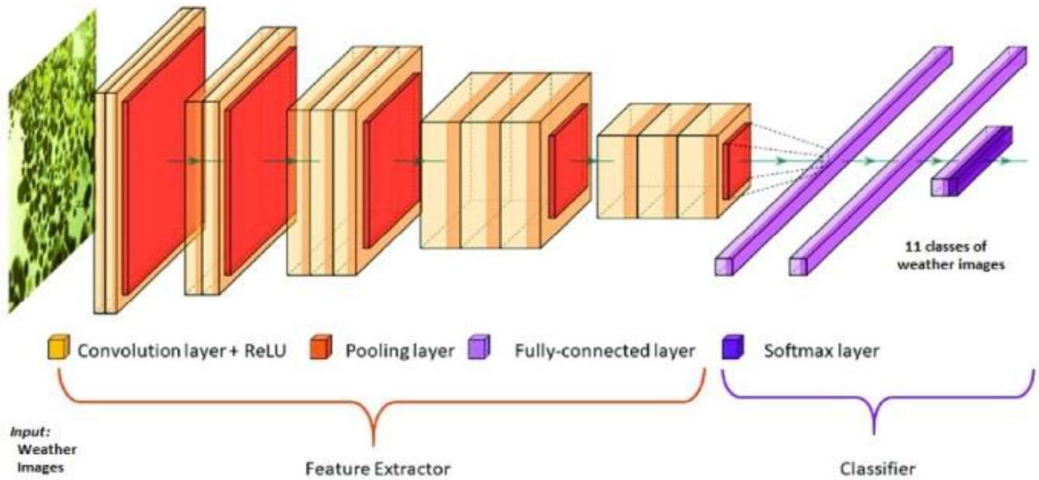

Fig. 2. CNN architecture for feature extraction and classification

At the second stage for the classifier, we employ XGBoost classifier to identify the weather images. In which, XGBoost algorithm stands for Extreme Gradient Boosting, a highly efficient machine learning algorithm based on a combination of techniques to adjust error weights on weaker models to create a stronger model. XGBoost algorithm principle is based on decision tree and gradient enhancement technique to give the optimal model. Sequentially generated new trees minimize the error from the previous tree by relearning the error of the previous tree, performing error correction to get a better tree. XGBoost was originally introduced by Chen and Guestrin (2016) to improve the performance and speed of decision trees according to the principle of gradient-boosted [9].

According to the description of the XGBoost algorithm given by authors of Chen and Guestrin [9], XGBoost works as follows:

For a given dataset with n samples and m features $D = \{(x_i, y_i)\}$ $(|D| = n, x_i \in R^m, y_i \in R)$, apply a model that combines the tree uses K enhancement functions to predict the output.

$$\hat{y}_i = \emptyset(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in F \quad (1)$$

where $F = \{f(x) = w_q(x)\}$ $(q : R^m \to T, w \in R^T)$ is the space of the regression tree (also known as CART). Here q is a representation for the structure of each tree, mapping a data sample to the corresponding leaf index. T is the number of leaves on the tree. Each f_k corresponds to an independent tree structure q and leaf weight w.

 ISBN: 978-604-80-8083-9

To find out the set of functions used in the model, the following normative objective function minimization algorithm:

$$\mathcal{L}(\emptyset) = \sum_{i=1}^{n} l(\hat{y}_i, \mathrm{y}_i) + \sum_{k}^{K} \Omega(f_k) \quad (2)$$

$$\text{where } \Omega(\mathrm{f}) = \gamma\mathrm{T} + \frac{1}{2}\lambda||\mathrm{w}||^2$$

Where, l is a differentiable convex loss function used to measure the difference between the predicted value $\hat{\mathbf{y}}_{\mathbf{i}}$ and the actual value y_i. The second component Ω is the penalty for model complexity (e.g. function of a regression tree). The additional normalization component smooth the learned final weights to avoid over-fitting. Visually, the normative objective tends to choose a model that uses simple but highly predictive functions.

The Gradient Tree Boosting algorithm is performed when the model is continuously trained in the way of feature addition. Formally, if $\hat{y}_i^{(t)}$ is the i-th prediction value at the t^{th} loop, the algorithm will need to add the f_t component to reduce the objective function as follows:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(\mathrm{y}_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \quad (3)$$

The second order approximation is used to optimize faster than the objective function in the algorithm implementation.

$$\mathcal{L}^{(t)} \cong \sum_{i=1}^{n} \left[l\left(\mathrm{y}_i, \hat{y}^{(t-1)} + g_i f_t(x_i)\right) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (4)$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l\left(\mathrm{y}_i, \hat{y}^{(t-1)}\right)$ và $h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(\mathrm{y}_i, \hat{y}^{(t-1)}\right)$ is the first and second order gradients on the loss function. We can remove the constants to obtain a simpler objective function as follows in step t.

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} [\mathrm{g}_i f_t(x_i) +] l\left(, \hat{y}_i^{(t-1)} + \frac{1}{2} h_i f_t^2(x_i)\right) + \Omega(f_t) \quad (5)$$

Definition $I_j = \{i|q(x_i) = j\}$ is the set representing the composition of leaf j. We can calculate the optimal weight w_j^* of leaf j by:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (6)$$

Calculate the corresponding optimal value by:

$$\tilde{\mathcal{L}}^{(t)} = -\frac{1}{2}\sum_{j=1}^{T}\frac{\left(\sum_{i\in I_j} g_i\right)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma\mathrm{T} \quad (7)$$

Equation (7) can be used as a scoring function to measure the quality of a tree structure q. This score is the same as the classification score for evaluating decision trees, except that it is computed for a wider range of objective functions.

In fact, the proven XGBoost algorithm optimizes speed and performance for building predictive models. At the same time, the XGBoost algorithm uses a variety of data formats, including tabular data of different sizes and layered data types. The algorithm of XGBoost is now widely by data scientists, is a scalable machine learning system for tree boosting which can help to avoid over-fitting. It performs well on its own and has been shown to be successful in many machine learning problems.

In our model, the XGBoost algorithm was implemented as the tree model is usually used as a primary classifier in XGBoost System. The features extracted from CNN were fed to train and test the XGBoost classifier in this study.

4 Results

4.1 Dataset

This study uses the dataset name as WEAPD [10] that contained 6,862 images. The authors have collected weather images from various sources. Figure 3 below depicts a few examples of images from this dataset.

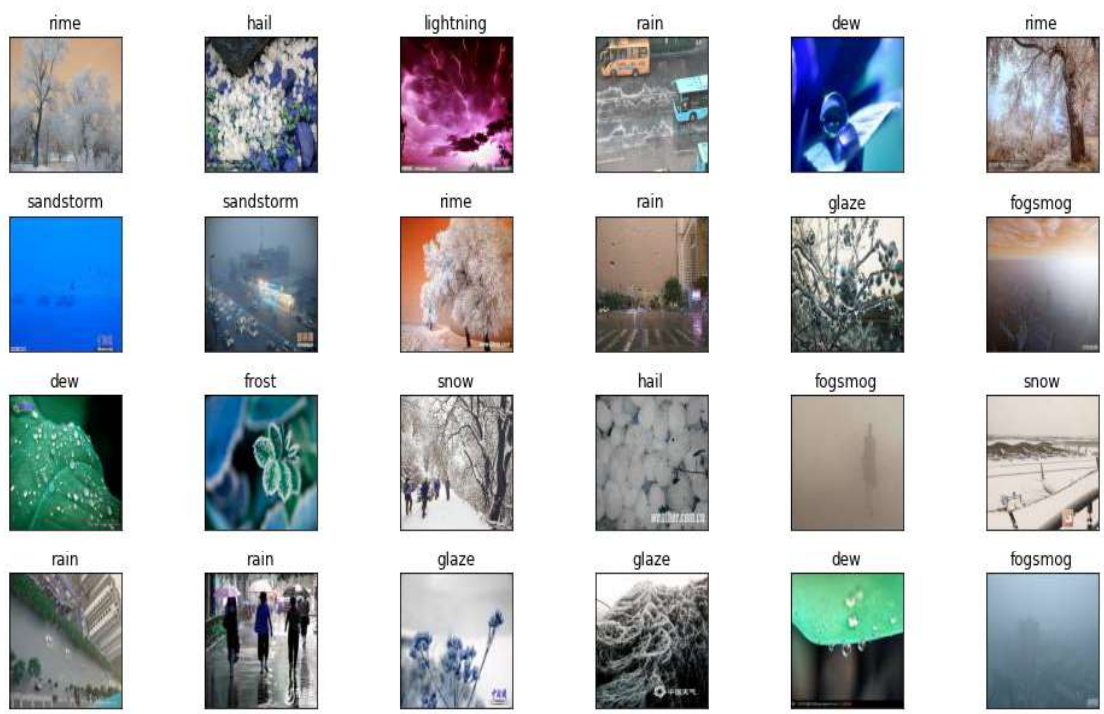

Fig 3. Example of weather images dataset

 ISBN: 978-604-80-8083-9

In this image dataset, the weather images were divided into 11 different image classifications (see Fig. 4). This set of 11 subclasses includes: dew, fogsmog, frost, glaze, hail, lightning, rain, rainbow, rime, sandstorm, and snow. We see that the dataset is not balanced data (imbalance images) among all the image data classes with the much higher number of rime images. But we keep the dataset as original for use, even though it may lead to lower accuracy.

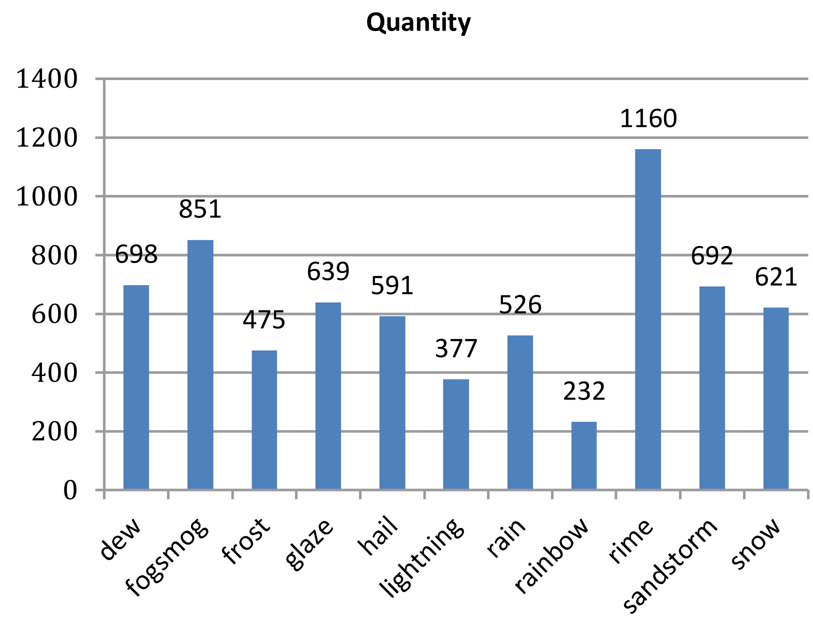

Fig. 4. Number of images by labels

As we can see that this dataset is highly imbalance due to too large number of images on rime. But we use this primitive dataset. Because, the aim of this study is to check whether what kind of hybrid deep learning has the best performance in the problem of weather image classification, not focus on accuracy of models in classification. The data is divided 80% for the training part, and 10% for the validation and 10% for the test of the model using the random splitter.

4.2 Experiments and results

First of all, we test the performance of a simple traditional Convolutional Neural Network to classify the weather images. Our simple CNN model has 3 Convolutional layers, the first with 128 filters, and the two remained layers with 64 nodes, each layer is followed by a Max Pooling and Dropout layers. The following layers are a Flatten layer and a Fully Connected layer with 128 nodes. The activation function, namely Rectified Linear Unit (ReLU) function is used after each convolutional layer. The optimizer is used namely Adam, and loss measures is categorical cross-entropy. In this simple traditional Convolutional Neural Network model, the Softmax function was employed to classify the weather images after the layer of Fully Connected. In the training process, we run model for 10 epochs with early stopping used to monitor the change of optimization.

Secondly, the experimental process removes the last layer of image classification of the above simple CNN (which used Softmax function), we keep only the feature extraction classes. And then, we use XGBoost to classify the weather images. We experimentally set the hyper-parameters of maximum tree depth equal 3, the minimum child weight valued at one, and the number of estimators takes value of 100 in XGBoost algorithm. In addition, the initial prediction score of all instances, global bias equal 0.5, and the model specified which booster to use booster is 'gbtree'. The subsample ratio of columns for each level is 1, and the subsample ratio of columns for each split is also set at 1. The learning rate for boosting is set at 0.1, and the regularization term on weights (xgb's lambda) is set at value of 1.

Thirdly, for the other comparative models, we kept the feature extraction classes at layers as the same as above experiment with XGBoost. But at this stage we test them with Support Vector Machine Classification, namely SVC classifier. The hyper parameter specified the kernel type to be used in the algorithm to be used by 'rbf', the degree is valued at 3 and the gamma chose 'scale'. The regularization parameter is equal at 1.0, and gamma is set at scale. In addition, we set the shrinking parameter is at true, the size of the kernel cache is at 200, class weight is none, decision function of shape is one-vs-rest.

In the next step, we take place to test with Decision Tree Classifier. The feature extraction classes were kept at layers of previous simple CNN model with 3 Convolutional layers we mentioned above. The hyper parameters for Decision Tree Classifier are as following values. The criterion is 'gini' for the Gini impurity, as function to measure the quality of a split. The minimum number of samples required to split an internal node is at 2 and the maximum depth of the tree are expanded until all leaves are pure or until all leaves contain less than 2 samples. The minimum number of samples is set at 1 training samples, required to be at a leaf node, it leaves at 1 training samples in each of the left and right branches. The minimum impurity decrease is at 0 to let a node would be split if this split induces a decrease of the impurity greater than or equal to 0. The class weight is set at none to let all classes which were supposed to have weight one.

Next, we test with AdaBoost classifier as the same as XGBoost and other above classifiers. The hyper parameters for AdaBoost classifier are followings. The base estimator is same as Decision Tree Classifier initialized with max depth equals 1. The maximum number of estimators at 50, we use the SAMME.R real boosting algorithm, Since there is a trade-off between the learning rate and maximum estimators parameters so we set learning rate at value of 1.0.

In next step, we test with Multi-layer Perceptron classifier as the same as XGBoost and other above classifiers. The hyper parameters for Multi-layer Perceptron classifier are hidden layer sizes that are equal 100, the activation function for the hidden layer is 'relu', the rectified linear unit function. The used weight optimization is 'adam', the initial learning rate used at 0.001; momentum for gradient descent update is at 0.9. The maximum number of iterations is at 200, and we do not use the early stopping to terminate training when validation score is not improving. The maximum number of epochs to not meet improvement is set at 10.

 ISBN: 978-604-80-8083-9

To sum up, we simply implement a traditional Convolutional Neural Network with 3 Convolutional layers for feature extraction. The classification is tested with Softmax function for categorical cross-entropy. Then, we used 4 other classifiers, namely XGBoost, SVC and Decision Tree Classifier, AdaBoost Classifier, Multi-layer Perceptron Classifier. The test results of CNN-Softmax, CNN-XGBoost, CNN-SVC, CNN-Decision Tree Classifier, CNN-AdaBoost, CNN-Multi-layer Perceptron Classifier models are shown in Table 1 below.

Table 1. Results on accuracy of models

Model	*Accuracy (%)*		
	Train	*Valid*	*Test*
CNN-Softmax	71.00	66.00	69.00
CNN-XGBoost	99.98	73.68	**72.75**
CNN-SVC	75.78	71.18	71.74
CNN-Decision Tree	99.98	66.47	65.94
CNN-AdaBoost	55.58	51.91	52.46
CNN-MLP	87.52	71.03	70.72

We can think that the accuracy is quite low (72.75%). But the research question of this study is not accuracy, it aims to check whether what kind of hybrid deep learning has the better performance compare to other hybrid models regarding the problem of weather image classification. Therefore, we implement a very simple CNN with 3 Convolutional layers for feature extraction which make lower accuracy but run faster. Our hypothesis does not focus on accuracy but performance of models in classification. Also, the dataset were imbalance acceptance which leads to lower accuracy. If we want higher accuracy, we should implement a transfer learning models, such as EfficientNet, DenseNet, ResNet, InceptionNet… which were deeply trained with million of images in ImageNet database. The aim of this study is to investigate the hybrid model, which one performs better in term of weather image classification problem. Therefore, the CNN model to make feature extraction is also simple; only 3 convolutional layers not include any batch normalization layers as our purpose is in order to speed up the process of experiments.

The table 1 above shows that the performance of CNN-XGBoost is the best among other 4 models in the problem of weather image classification. Its values of accuracy on test set at 72.75% which is higher than other comparative models on the same tested dataset of weather images.

5 Conclusion

In this study, there were four hybrid models and a simple CNN models, totally five models, were employed them on the same weather image classification problem. We used a simple CNN model with 3 convolutional layers as feature extractor and then they are built on the same network architecture with respective different classifiers. All of them are employed in the same training, validation and test set of weather images. The outcomes of experiments show that the CNN-XGBoost has the best performance among other 5 models in the problem of weather image classification.

References

1. Xiao, H., Zhang, F., Shen, Z., Wu, K., & Zhang, J.: Classification of weather phenomenon from images by using deep convolutional neural network. Earth and Space Science, 8, https://doi.org/10.1029/2020EA001604 (2021).
2. Mohammad F. N. & Selvia F. K.: Weather image classification using convolutional neural network with transfer learning. AIP Conference Proceedings, Volume 2470, Issue 1, https://doi.org/10.1063/5.0080195 (2022).
3. Mohamed R. I, James H. and Tao C.: WeatherNet: Recognising Weather and Visual Conditions from Street-Level Images Using Deep Residual Learning. International Journal of Geo-Information, ISSN: 2220-9964, ISPRS - International Society for Photogrammetry and Remote Sensing (2019).
4. Khan, M.N.; Ahmed, M.M.: Weather and surface condition detection based on road-side webcams: Application of pre-trained convolutional neural network. Int. J. Transp. Sci. Technol. vol. 11, pp. 468-483 (2021).
5. Minhas S, Khanam Z, Ehsan S, McDonald-Maier K, Hernández-Sabaté A.: Weather Classification by Utilizing Synthetic Data. 22(9):3193. doi: 10.3390/s22093193. PMID: 35590881; (2022).
6. Thongsuwan S., Jaiyen S., Padcharoen A., Agarwal P.: ConvXGB: A new deep learning model for classification problems based on CNN and XGBoost. Nuclear Engineering and Technology, ISSN 1738-5733, Vol 53, Issue 2, Page: 522-531 (2021).
7. Thiyagarajan Sahana: Performance Comparison of Hybrid CNN-SVM and CNN-XGBoost models in Concrete Crack Detection. Theses Masters, Technological University Dublin, https://arrow.tudublin.ie/scschcomdis/195/ (2019).
8. Huang, G-L, He, J, Xu, Z, Huang, G.: A combination model based on transfer learning for waste classification, Concurrency Computat Pract Exper, https://doi.org/10.1002/cpe.5751 (2020).
9. Chen and Guestrin: XGBoost: A Scalable Tree Boosting System. KDD'16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 785–794, https://doi.org/10.1145 /2939672.2939785 (2016).
10. Xiao, Haixia, "Weather phenomenon database (WEAPD)", https://doi.org/10.7910 /DVN/M8JQCR, Harvard Dataverse, V1 (2021).

 ISBN: 978-604-80-8083-9