EVALUATION OF VISION TRANSFORMER ON WEATHER IMAGE RECOGNITION

Phi Cong Huy[1], Tran Quy Nam[2*]

Abstract – *This study implements a Vision Transformer 16x16 Words model for weather image classification. Its performance is compared with other traditional convolutional neural network architectures, namely EfficientNetB2, DenseNet201, EfficientNetB7, and MobileNetV2. These models are implemented by transfer learning techniques for the classification of images. In order to ensure comparative performance, the same hyper-parameters of their models, such as dropout rate, optimizer, and learning rate are employed identically. Furthermore, the same dataset on weather image phenomena is applied to all those models with the same training, validation, and testing dataset of weather image classification. The dataset of 11 different image classes that are collected from different resources of weather images with various kinds of weather phenomena is employed. The test results of performance show that the Vision Transformer gives the best results at 86.20%, which is suitable for application in evaluating weather image classification problems.*

Keywords: convolutional neutral network (CNN), image, Vision Transformer (ViT), weather.

I. INTRODUCTION

In our daily lives, weather information always plays a very important role in all aspects of human activities. Nowadays, socioeconomic development and global warming also leads to fast-changing weather conditions. Weather knowledge always has a large impact on human life and the socio-economic development of many countries in the world. The correct recognition of weather phenomena is one of the important factors to support our lives and study nature's development. There are some ways to recognize weather phenomena, such as measurement of temperature, atmosphere, observational data collected by Doppler radar, weather satellites, and other instruments such as weather balloons to measure atmospheric parameters [1]. The weather models use mathematical and statistical equations, along with new and past weather data, to provide informative guidance. In computer science, the development of computer vision systems has achieved great success in many areas, such as highly accurate image processing, which has already led to many applications in surveillance, navigation, and driver assistance systems.

The automatic methodological solutions of weather image classification thanks to AI technology can help people to obtain sustainable progress and development [2]. The processing or identification of weather images that are taken from drones or cameras is an important method in weather forecasting, environmental assessments, and warning of dangerous transportation. In terms of environmental assessments, it is important to classify the respective weather phenomenon to alarm people before going outside, which provides appropriate guidance to people regarding attire and travel plans by the prevailing weather conditions. In addition, the highly accurate recognition of weather images can help people avoid the negative effects or damages of natural disasters.

For weather forecasting, correct recognition of images of past weather phenomena affects the ability to predict future weather conditions. Thus, effective weather forecasting and classification

[1,2]Posts and Telecommunications Institute of Technology, Vietnam

*Corresponding author: namtq@pit.edu.vn

Received date: 17th July 2023; Revised date: 30th August 2023; Accepted date: 30th August 2023

leads to more exact assessment of environmental quality and more positive effects on agriculture, as 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, effectively recognizing and classifying weather images is a significant issue in our daily lives.

In this paper, we implement transfer learning with pre-trained models to test the performance of convolutional neural network (CNN) and Vision Transformer on a weather image phenomenon classification. We implemented four traditional CNN architectures, namely EfficientNetB2, DenseNet201, EfficientNetB7, and MobileNetV2, and another model in addition to Vision Transformer (ViT_B16) 16x16 Words for weather image phenomenon classification.

II. LITERATURE REVIEW

Over the world, there have been many studies that use the techniques of machine learning models or deep learning models to recognize weather images. Xiao et al. [3] implemented a CNN that was named MeteCNN for weather phenomena classification and their model provided good results. MeteCNN used VGG16 as the framework to build the proposed MeteCNN model. The MeteCNN added a global average pooling layer instead of the max pooling layer before the softmax layer for the classification task. Mohammad et al. [4] studied the set of weather images to classify them using CNN with Transfer Learning. Their model comprised four pre-trained CNN architectures, namely MobileNetV2, VGG16, DenseNet201, and Xception to predict classes of weather images. They used the method of transfer learning which aimed to promote the speed of training models to get better and faster performance. They applied those four pre-trained CNN architectures to the weather images. Their dataset comprised six classes, named as classes of cloud, rain, sun-shining, sunrise, snow, and fog as weather labels. The outcomes of their research showed that the Xception has the best accurate number at 90.21%, meanwhile, the pre-trained CNN of MobileNetV2 has an accuracy of 83.51%.

Mohamed et al. [5] introduced a neural network for the identification of street images, in which they did not use any pre-defined linkages in the images. They designed a CNN model and called it WeatherNet, which was not new but it was based on ResNet50 architecture. The WeatherNet model tried to get features of images of weather to classify the time of weather, such as dawn-time or dusk-time, daytime or night-time, and their model tried to classify the classes of weather images, dividing them into four classes of weather, namely clear-sky, rain, snow, and fog to reflect real weather conditions. Their WeatherNet showed very good results on weather multimedia datasets, such as images or video. Khan et al. [6] studied some CNN models that also tried to classify three weather kinds of conditions. They were namely clear-sky, light-snow, and heavy-snow. They also tried to recognize the surface of places in weather conditions, including dry-surface, snowy-surface, and wet-surface. They applied them to several pre-trained CNN-based models, which comprised AlexNet, GoogLeNet, and ResNet18 by techniques of transfer learning on those pre-trained models. The tests on the real datasets showed that the best ResNet18 performed best among all tested models. ResNet18 produced the highest number of accuracy rate, reaching 97% for weather image classification. Minhas et al. [7] studied weather conditions in the future from real-world images via targeting classification using neural networks. In their article, the results indicated that the use of hybrid datasets in connection with the exacted real datasets in the world could help to increase the productivity of training time of the CNNs by 74%.

Kang et al. [8] presented a CNN-based network that was a kind of inception and learned from GoogLeNet architecture. They researched two classes of weather conditions that included rainy and snowy. Their models tried to classify a lot of input images and separate them into two categories of rainy or snowy or neither, such as sunny, for example. The authors made two full checks on weather image datasets. The first one was implemented with the stage of preprocessing images, and the second one was conducted without the stage of preprocessing images. Both checks were tried to be evaluated by their performances the proposed method was based on GoogLeNet architecture and both approaches showed very good results on the tested dataset. Tran-Trung et al. [9] conducted research to identify the status of clouds that were diversified by their volume size, shape-draw, thick-thin size, height, and coverage size. They considered the color characteristics, extracted mainly from four features mentioned above, and put those features for classifying the cloud images. The outcomes of the authors' paper showed that their proposal provided a good result compared to other techniques, for example, histogram choice.

Horv'ath et al. [10] proposed a technique, in which they used a Vision Transformer based on unsupervised learning to identify the images collected from the satellite. They tested the dataset of images collected from a satellite that also consisted of some spliced items. They concluded that their implementation of Vision Transformer performed better than other current unsupervised techniques on splicing items. Li et al. [11] implemented a model that combined a vision transformer and augmentation of images on the weather image classification problem. The research aimed to resolve the problems that lack the capability for feature extraction resulting from traditional deep learning models. They also hypothesized that Vision Transformer could resolve the problem of a low number of classes of weather phenomena in the weather image dataset. Finally, their research showed that the classified accuracy which was given by their tested Vision Transformer was much higher than the one given by other traditional deep learning models.

III. RESEARCH METHODS

This study implements five image recognition models which are applied for weather image classification. They are namely Vision Transformer, DenseNet201, MobileNetV2 and EfficientNetB2, EfficientNetB7. The transfer learning with pre-trained models was employed to test their performance. The same dataset was trained, validated, and tested by each model on the identical dataset to compare the performance of each model on the problem of weather image classification. The following paragraphs discuss in detail the architecture, description, running principle, and mechanism of those five testing models.

A. Vision Transformer (ViT)

In 2021, Dosovitskiy et al. [12] proposed a new approach in computer vision problems, inspired by the application of Transformer architecture to natural language processing. The group of researchers from Google Research introduced the Vision Transformer architecture (see Figure 1) which was a Transformer architecture version for the image. This architecture has achieved a lot of outstanding results in many different problems. The outcomes from research and application of the transformer model in natural language processing have become very impressive. However, in terms of computer vision, the application and research of the transformer model is still limited and lacks research. In computer science, when encountering computer vision problems such as classification, image recognition, object detection, and object segmentation, the convolutional architecture, namely the CNN network is still the familiar architecture that often being used.

Transformer architecture uses the mechanisms of the attention mechanism that carefully considers the importance of each area of input data. Transformer in machine learning includes many layers of the self-attention mechanism, mainly used in the AI fields of natural language processing and computer vision. The Transformer model in machine learning expresses the potential way of a generalized learning method. The transformer can be applied to different datasets in computer vision, to achieve modern accuracy with fewer parameters in the landscape of limited computing resources.

The traditional transformer architecture takes as input a string of 1D embedding tokens. Therefore, to process the input as a 2D image, the Vision Transformer model breaks down the input image into fixed-sized packets (patches) like the word embedding sequences used in the traditional transformer model that are used for text.

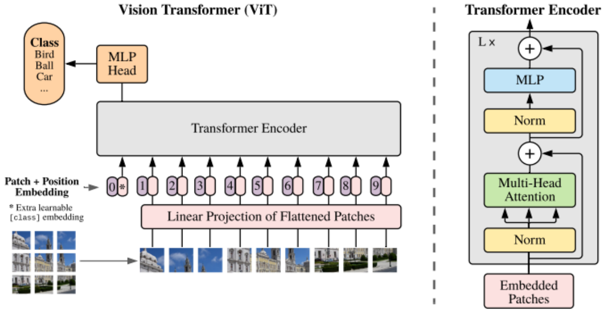


Fig. 1: Vision Transformer architecture with attention mechanism [12]

The transformer encoder takes an input as a combination of patch embedding and position embedding information, including Multihead Attention classes, multi-layer perceptron (MLP) blocks, and Layer norm. In the paper, Dosovitskiy et al. [12] mentioned two characteristics of Vision Transformer, namely inductive bias and associative architecture. Regarding the inductive bias, the Vision Transformer has less inductive bias in a particular image than the CNN architecture. The reason is that in the CNN network architecture, the three characteristics of localization, 2-way neighbor architecture and equivalent displacement are presented in each layer. Meanwhile, in Vision Transformer, only the MLP layer has equivalent localization and displacement properties. In terms of associative architecture, instead of taking the image as input directly, Vision Transformer is often combined with CNN architecture to extract features from the input image and then take the final feature map as input to the ViT model.

B. DenseNet201

In 2017, a research group of Huang et al. [13] proposed a DenseNet 201 network based on CNN. The key idea is that the convolutional network analysis can be deeper, more accurate, and more efficient for training if it contains shorter connections between the input layers and the output layers. The DenseNet network has a structure consisting of dense connection blocks (dense blocks) and subsequent layers of connectivity (transition layers), as seen in Figure 3. In the traditional CNN architecture, if there are L layers, there are L connections, in DenseNet there will be L(L+1)/2 connection.

The idea of DenseNet works on the following principle: see x_0 as a single image that was fed via a convolutional network. This neural network consists of several of L layers, each layer using a nonlinear transform $H_\ell(.)$ in which ℓ is the class index. $H_\ell(.)$ can be a set normalization function (Batch Normalization - BN), a ReLU function, a Pooling function or a convolution. The class output symbol ℓ_{th} is x_ℓ then:

For the ResNet network, traditional convolutional networks forward the data input source connected to the output of the ℓ_{th} layer as input to the $(\ell+1)_{th}$ sub-layer, resulting in an increase in layer shifting: $x_\ell = H_\ell(x_{\ell-1})$. ResNets adds a short-cut connection to skip un-linear transformations with an identity function:

$$x_\ell = H_\ell(x_{\ell-1}) + xx_{\ell-1} \qquad (1)$$

For densely connected DenseNet networks, to further improve the information between layers, Huang et al. [11] propose another connection: a direct connection from each layer toward all next following layers. Figure 2 below describes the connection diagram of that DenseNet network. As a result, the ℓ_{th} layer receives the characteristics and features that come from previous layers, x_0, x_1, … $x_{\ell-1}$, that become its input:

$$x_\ell = H_\ell([x_0, x_1, \ldots x_{\ell-1}, x_{\ell-1}]) \qquad (2)$$

where, [x_0, x_1, … $x_{\ell-1}$] represents the combination of characteristics and features created in layers 0, 1, … $\ell - 1$. Because it is densely connected, Huang et al. [13] named this network architecture the densely connected network (DenseNet).

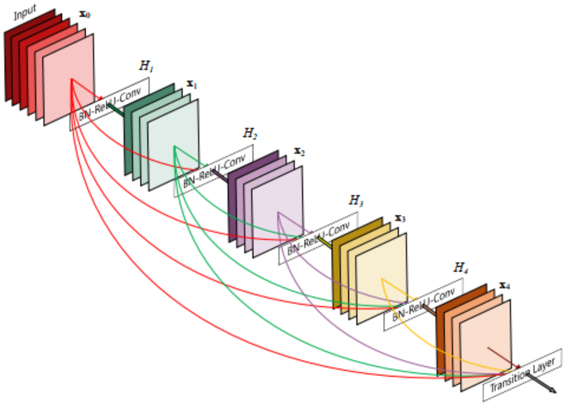
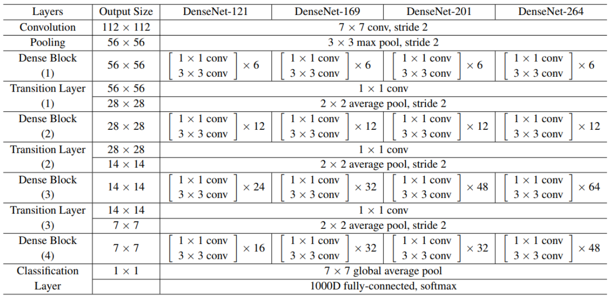


Fig. 2: DenseNet 201 network architecture with 5 densely connected blocks [11]

The DenseNet network differs from the ResNet network in that it does not perform the addition directly. Instead, the outputs of each mapping of the same length and width will be connected into a dense block in depth.



Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Fig. 3: DenseNet201 configuration parameters [13]

C. *EfficientNetB2 and EfficientNetB7*

In 2019, research published by Tan et al. [14] proposed an EfficientNet network based on CNN. The key idea is to recognize that CNN models achieve a positive and certain result with a fixed amount of computing resources. Therefore, to increase accuracy, models usually have the following three directions: increasing the depth of the model; increasing the width of each layer in the model; and improving the quality of the input (increasing image quality, and size). However, the ways to scale the model are mostly to choose one of these three directions above. The paper presented by Tan et al. [14] gave a new approach in balance to scale the CNN model to get better accuracy with fewer parameters and increased FLOPS (number of floating points to calculate per second).

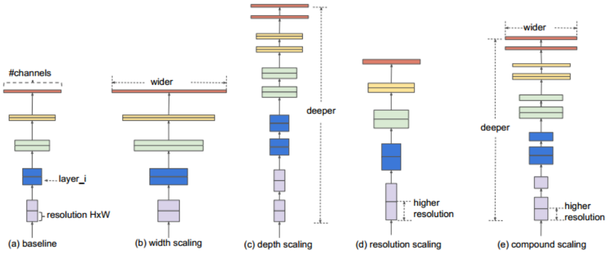


Fig. 4: Model scaling [6]

In Figure 4, graph (a) depicts a basic line network, and graphs (b) to (d) are the conventional connection lines that can go up only one dimension, including width, depth, and resolution. Graph (e) is the Tan et al. [14] proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio. In addition, the authors also presented the EfficientNet model B0 to B7, in which B0 is the base and B1-B7 is the tuning of B0. Their paper proposed a new expanding or scaling solution. This makes scaling all dimensions that may be depth width or resolution. The method used a simple but highly effective network architecture with compound coefficients. They expanded the effectiveness of the methodology by expanding the pre-trained models of MobileNets and ResNet.

They use the frame of a neural network to create a new basic line and make scaling to get a series of models, named EfficientNets. In practice, their EfficientNet-B7 got a high SOTA at 84.30% top-1 accurate rate on a dataset of ImageNet, meanwhile smaller and faster than other CNN networks (see Figure 5).

D. MobileNetV2

In 2017, MobileNet was introduced by a team of Google engineers in CVPR 2017 in their paper titled 'MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications' that was given by Andrew et al. [15]. The main strength that made MobileNet a highly accurate model was the low requirement of computing that lies in the improvement of the normal convolutional layer. In the network of MobileNet, two convolutional layers are employed, including SeparableConv and DepthwiseConv. The layer named SeparableConv will perform depth-wise a part of (spatial) convolution followed by pointwise convolution (see Figure 6). DepthwiseConv will only perform depth-wise spatial convolution (not counting point-wise convolution). Splitting the convolution thus largely decreases the computing cost and the size of their parameters in the network. MobileNet proposed a depth-wise separable convolution. The architecture shrinks the neural network model so that it can work on the limited resources on mobile devices. MobileNets are small, low-latency, low-power models parameterized to meet the resource constraints of a variety of use cases.

The new solution which was implemented inside MobileNet is to change the high computing cost of convolutional layers with depthwise separable convolutional units where each unit includes a 3x3 deep smart convolutional layer to filter the input, followed by a 1x1 point convolution layer combining these values that were filtered to make new features.

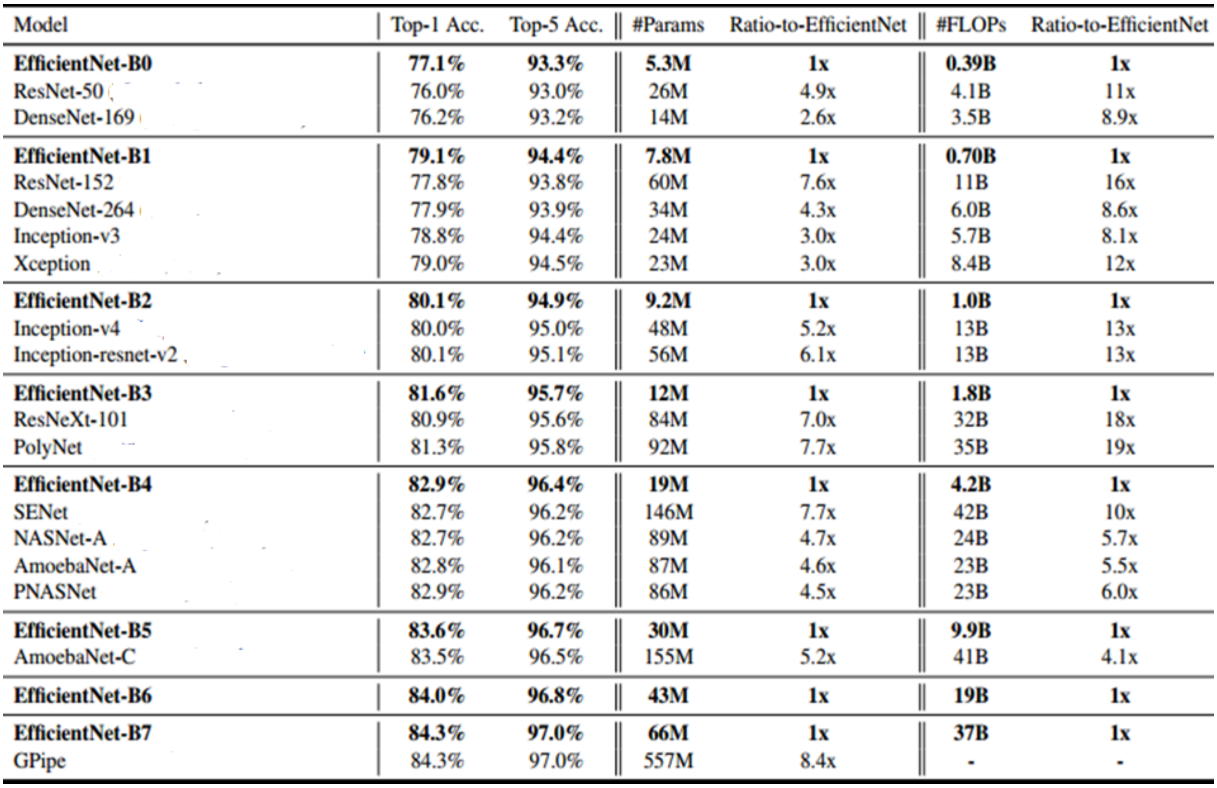
The MobileNet V1 architecture begins with a regular 3×3 convolution and is followed by 13 depth-wise separable convolution units. Inside the architecture of MobileNet V2, per unit consists of a 1x1 expanding layer in further addition to the depth and point convolution layers. It is different from V1, as the pointwise convolution layer of V2 is called a projection layer which projects data from a high number of channels into a tensor with a largely lower number of channels. The 1x1 batch expanding convolution layer will increase the number of channels based on the data expansion factor before diving into smart convolution. The second new feature of MobileNet V2 is residual connectivity. The remaining connection exists to support gradual flow on the network. Every layer inside MobileNet V2 also has the batch normalization function and ReLu functioning with activation form. Nevertheless, the output of the projected layer does not have a trigger function. The complete architecture of MobileNet V2 consists of 17 consecutive congested residual blocks, followed by a regular 1x1 convolution, a global average pooling layer, and a classifying layer. MobileNet V2 improves the performance of mobile device models on a variety of tasks and tests and a variety of sample sizes.

Mobilenet V2 is an improved version of Mobilenet V1. Its prominent optimization is to utilize the network structure, making it more effective and precise. MobileNet-v2 is a CNN with a depth of 53 layers. MobileNetV2 is a very big development over MobileNetV1, in which MobileNetV2 is based on the primary foundation originated from MobileNetV1. In which, they make use of depth-wise separable convolution as effective constructive units. In their proposal, V2 introduced two new characteristics to the framework of the network: 1) linear congestion between layers and 2) shortened connections between bottlenecks. The MobileNetV2 architecture contains the first fully convolutional layer with 32 filters, followed by the remaining 19 cluttered layers.

IV. RESULTS AND DISCUSSION

A. Dataset

This study uses the dataset named WEAPD [16], which contains 6,862 images.



Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPs	Ratio-to-EfficientNet
EfficientNet-B0	**77.1%**	**93.3%**	**5.3M**	**1x**	**0.39B**	**1x**
ResNet-50	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	**79.1%**	**94.4%**	**7.8M**	**1x**	**0.70B**	**1x**
ResNet-152	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	**80.1%**	**94.9%**	**9.2M**	**1x**	**1.0B**	**1x**
Inception-v4	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	**81.6%**	**95.7%**	**12M**	**1x**	**1.8B**	**1x**
ResNeXt-101	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	**82.9%**	**96.4%**	**19M**	**1x**	**4.2B**	**1x**
SENet	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	**83.6%**	**96.7%**	**30M**	**1x**	**9.9B**	**1x**
AmoebaNet-C	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	**84.0%**	**96.8%**	**43M**	**1x**	**19B**	**1x**
EfficientNet-B7	**84.3%**	**97.0%**	**66M**	**1x**	**37B**	**1x**
GPipe	84.3%	97.0%	557M	8.4x	-	-

Fig. 5: EfficientNet performance results on ImageNet [14]

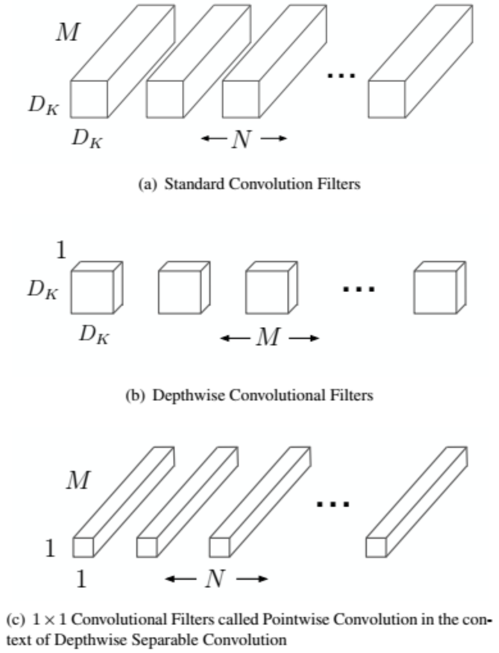
The authors have collected weather images from various sources. Figure 7 below describes several images taken from this dataset.

Inside this image dataset, the weather images were classified into 11 different image classes with respective quantities of images (see Figure 8). This set of 11 subclasses includes dew, fog smog, frost, glaze, hail, lightning, rain, rainbow, rime, sandstorm, and snow. In the distribution of the dataset in Figure 8, this dataset is not balanced data, indicating imbalanced image classes. There are a larger number of rime class images. However, the dataset is retained in its original form for usage, despite the potential impact on overall accuracy.

Despite that, the dataset was highly imbalanced due to a bigger number of images on the class of rime, but the dataset was used to check whether the ensemble deep learning model can have a good performance compared to separate pre-trained models when classifying weather images.

B. Experiments and results

In the first step, the dataset was organized into 80% images to be the training set, with the addition of 10% for the valid set and 10% for the testing set. The model ran with a random splitter of the respective division of the respective image dataset. The size of weather images is resized at 224 x 224 dimensions. Then, the study employs transfer learning with pre-trained models to test the accuracy of the above weather image dataset. All the models for implementation, namely EfficientNetB2, Vision Transformer (ViT_B16), DenseNet201, EfficientNetB7, and MobileNetV2 are all frozen previous layers except the last layers for classification. The same hyper-parameters for all five models are also set up similarly for all tested models for appropriate comparison. In which, the loss function as cross



(a) Standard Convolution Filters

(b) Depthwise Convolutional Filters

(c) 1 × 1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Fig. 6: SeparableConv and DepthwiseConv architecture [15]

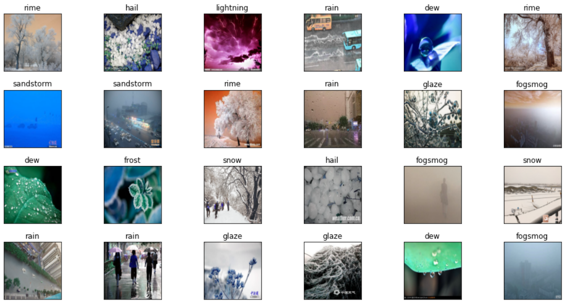


Fig. 7: Few weather images [16]

entropy loss is set, the dropout rate is set at 0.2, the optimizer is Adam, and the learning rate equals 0.001. The measurement of the accuracy rate of those five models on the same weather image dataset and the same hyper-parameters is depicted in Table 1 below.

Table 1: Performance on accuracy of respective models

Model	Train Loss	Train Acc.	Valid Loss	Valid Acc.	Test Acc.
EfficientNetB2	0.7130	0.8143	0.6944	0.8375	0.8458
VIT_B16	0.4603	0.8768	0.3939	0.8875	**0.8620**
DenseNet201	0.6857	0.8063	0.5010	0.8250	0.8538
EfficientNetB7	0.7613	0.8125	0.5463	0.8688	0.8350
MobileNetV2	0.7090	0.8196	0.6358	0.8438	0.8274

In Table 1, the test accuracy of Vision Transformer with 16x16 Words is the highest value at 86.20%. The second best accuracy is the model of DenseNet201 with 85.38% and the other models are relatively not too far behind the accuracy, ranging around 82% to 84%. Therefore, the accuracy of Vision Transformer is the best among those five models during implementation on the weather image dataset.

V. CONCLUSION AND RECOMMENDATIONS

This study employed the transfer learning on pre-train models for implementation, namely EfficientNetB2, Vision Transformer (ViT_B16), DenseNet201, EfficientNetB7, and MobileNetV2 on the weather images classification problem. The study used four pre-trained models of traditional CNN architectures, which were named EfficientNetB2, DenseNet201, EfficientNetB7, and MobileNetV2, and in addition for comparison with Vision Transformer (ViT_B16) 16x16 Words. Those models are all similarly implemented with a training set, valid set, and tested set when classifying images of weather phenomenon. Besides, this research keeps the same hyper-parameters, such as dropout rate, optimizers, and learning rate to make a comparison of the accuracy of the model. The outcomes of the mentioned experiments show that the Vision Transformer (ViT_B16) has the best performance at 86.20% compared with other models in the same dataset and the same problem of weather image identification. Future researchers may employ the Vision Transformer model with various kinds of image datasets. Furthermore, researchers should consider comparing Vision Transformer with other deep learning techniques, such as ensemble models, stacked models, and combined hybrid models with boosting algorithms by using GAN (Generative Adversarial Networks).

REFERENCES

[1] National Weather Service (NWS). *Advances in radars and satellites*. https://www.weather.gov/mqt/fitz_remote [Accessed 12th April 2023].

[2] Jaseena KU, Kovoor BC. Deterministic weather forecasting models based on intelligent predictors: A survey. *Journal of King Saud University-Computer and Information Sciences*. 2022;34(6): 3393–3412. https://doi.org/10.1016/j.jksuci.2020.09.009.

[3] 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*. 2021;8(5): 1–9. https://doi.org/10.1029/2020EA001604.

[4] Naufal MF, Kusuma SF. Weather image classification using convolutional neural network with transfer learning. In: Parung J, Pah ND, Sutrisna PD, Suciadi MFS. (eds.) *International conference on informatics, technology, and engineering 2021 (InCITE 2021): Leveraging smart engineering, 25-26 August 2021, Surabaya, Indonesia*. New York, United States: AIP Publishing; 2022. https://doi.org/10.1063/5.0080195.

[5] Ibrahim MR, Haworth J, Cheng T. WeatherNet: Recognising weather and visual conditions from street-level images using deep residual learning. *ISPRS International Journal of Geo-Information*. 2019;8(12): 549. https://doi.org/10.3390/ijgi8120549.

[6] Khan MN, Ahmed MM. Weather and surface condition detection based on road-side webcams: Application of pre-trained convolutional neural network. *International Journal of Transportation Science and Technology*. 2022;11(3): 468–483.

[7] Minhas S, Khanam Z, Ehsan S, McDonald-Maier K, Hernández-Sabaté A. Weather classification by utilizing synthetic data. *Sensors*. 2022;22(9): 3193. https://doi.org/10.3390/s22093193.

[8] Kang LW, Feng TZ, Fu RH. Inception network-based weather image classification with pre-filtering process. In: *23rd International Computer Symposium*. Singapore: Springer Singapore; 2018. p.368–375. https://doi.org/10.1007/978-981-13-9190-3_38 (2019).

[9] Kiet Tran-Trung, Ha Duong Thi Hong , Vinh Truong Hoang. Weather forecast based on color cloud image recognition under the combination of local image descriptor and histogram selection. *Electronics*. 2022;11(21): 3460. https://doi.org/10.3390 /electronics11213460.

[10] Horváth J, Baireddy S, Hao H, Montserrat DM, Delp EJ. Manipulation detection in satellite images using vision transformer. In: *2021 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW), Nashville, TN, USA*. IEEE Xplore; 2021. p.1032-1041. https://doi.org/10.1109/CVPRW53098.2021.00114.

[11] Li J, Luo X. A study of weather-image classification combining vit and a dual enhanced-attention module. *Electronics*. 2023;12(5): 1213. https://doi.org/10.3390/electronics12051213.

[12] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. *An image is worth 16x16 words: Transformers for image recognition at scale*. To be published in ICLR 2021. *arXiv*. [Preprint] 2021. Version 2. https://doi.org/10.48550/arXiv.2010.11929.

[13] Huang G, Liu Z, Van DML, Weinberger KQ. Densely Connected Convolutional Networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*. IEEE Xplore; 2017. p.2261-2269. doi: 10.1109/CVPR.2017.243.

[14] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. To be published in PMLR 97. *arXiv*. [Preprint] 2020. Version 5. https://doi.org/10.48550/arXiv.1905.11946.

[15] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *Computer Vision and Pattern Recognition*. arXiv; 2017. https://doi.org/10.48550 /arXiv.1704.04861.

[16] Xiao H. *Weather phenomenon database (WEAPD)*. V1. Harvard Dataverse; 2021. https://doi.org/10.7910/DVN/M8JQCR.

Creative Commons Attribution-NonCommercial 4.0 International License.