Farhan Mohammed Abdul Qadir, Hamzah Luqman

Abstract

I. Introduction

Throughout human history, weather has played an integral role in shaping our lifestyles across different parts of the world, both as a boon and a detriment to our livelihood. Rain invigorates the land and brings about crops, yet it can cause floods and consequently, loss of life. Sunny days are usually the time for recreation and work, while hail and sandstorms are too disruptive and dangerous for day-to-day operations. Weather patterns like these greatly influence our schedules each day, and thus having an educated foresight into the weather in the near future can not only prevent needless injury, but also give insight to schools and workplaces on whether to continue operating or not. With the rise of turbulent climate patterns compounded with an increase of human activity in recent times, the significance of weather detection has become very apparent for environmental safety and climate change observation.

The Middle East, namely the Kingdom of Saudi Arabia, is home to a usually humid and hot climate. Rarely is it rainy or windy except for a month or two near the end of the year. It is this infrequent change in weather conditions that catches the public by surprise in events of sudden rain, strong winds, and sandstorms. These scenarios not only abruptly jumble school and employee schedules but also pose a threat to traffic safety due to lack of coordination and sophisticated sewage systems. Moreover, human-dependent forecasts may either take too long or may not be foreseen precisely due to subtle signs and sudden changes in weather patterns. In order to complement such shortcomings and bolster detection rates, we propose to implement deep learning techniques in the form of attention-based vision transformers to recognize weather efficiently and accurately.

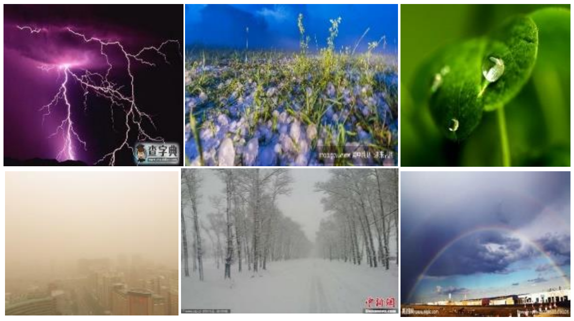


Fig 1. Samples from the dataset representing (from top left to bottom right) dew, hail, lightning, sandstorm, snow, and rainbow.

II. Problem description

There has been a steady surge of interest on weather recognition through computer vision in recent years, with convoluted neural networks being among the most popular implementations. Elhoseiny et al. (2015) trained a CNN consisting of 5 convolution/pooling layers and 3 fully connected layers for binary classification of images of the Weather Database (‘cloudy’ and ‘sunny’). Using an 80-20 training/testing split, the model achieved an 82.2% average test accuracy on 2,000 images. The same dataset was the subject of another study conducted by An et al. (2018), whose combination of AlexNet, ResNet, and a multi-class SVM attained a 90% average test accuracy for binary classification. Their model performed even better on 4-class classification however, achieving an approximate accuracy of 98-99%. Al-Haija et al. (2020) achieved similar results using a ResNet18 model with pretrained ImageNet weights, observing a 98.22% test accuracy on the ‘multi-class weather recognition dataset’ which consisted of 4 classes: ‘sunrise’, ‘shine’, ‘rain’, and ‘cloudy’. Ibrahim et al. (2019) extracted features of weather images using the ‘WeatherNet’ model, a CNN parallelly comprised of four modified ResNet50 models. Data augmentation and sampling techniques were applied onto the Image2Weather and Multi-class Weather Image datasets to ensure balanced representations of both daytime and nighttime images. The strong feature extraction abilities of the model enabled it to achieve a precision of 92.4% on their test set when differentiating between ‘Sunny’, ‘Cloudy’, ‘Rainy’, and ‘Foggy’ classes. A deeper level of weather classification was conducted by Xiao et al. (2021), who developed the ‘MeteCNN’, a VGG16-inspired CNN model, to train on images of 11 classes: ‘Hail’, ‘Rainbow’, ‘Snow’, ‘Lighting’, ‘Dew’, ‘Sandstorm’, ‘Frost’, ‘Smog’, ‘Rime’, and ‘Glaze’. Omitting the VGG16’s fully connected layers in favor of a global average pooling layer among other modifications, the MeteCNN model attained the highest precision rates for ‘Hail’, ‘Rainbow’, ‘lightning’, and ‘dew’ from 97-100% while performing the weakest for ‘Snow’ and ‘Glaze’ at 85% precision. The model achieved an average accuracy of 92.68%, surpassing established models like ResNet34, MobileNet, and VGG19.

III. Dataset

The ‘Weather Phenomenon Database’, published as part of the Harvard Dataverse by Xiao H. et al. (2021), is a compilation of 6,862 high-resolution images of varying dimensions, representing 11 different types of weather: dew, fog, frost, glaze, hail, lightning, rain, rainbow, rime, sandstorm, and snow. Since some of the represented types of weather were not relevant to Middle Eastern weather and are generally not known as common phenomena, we decided to limit the scope of classification to 7 weather conditions common throughout the region: dew, hail, lightning, rainbow, sandstorm, and snow. The narrowing of classes resulted in a dataset consisting of 3,737 samples.

IV. Methodology

The primary deep learning technique that we worked with was Transfer Learning of Convolutional Neural Networks (CNNs). Transfer learning is a technique of re-using a model's learned parameters/weights that has already been trained on a huge dataset and then fine tuning its layers accordingly to suffice the classification problem at hand.

Three state-of-the-art models in the field of computer vision such as Resnet52/152, MobilenetV2, and VGG16 were employed to classify 7 classes of the WEAPD. Furthermore, to compare our results in a systematic way, we constructed a custom baseline model that mainly consisted of CNN-BN layers. However, the main highlight of our paper is the full utilization and fine-tuning of a vision attention module called Vision Transformer or ViT that paved its way recently in the computer vision literature. The description of all the models that we used is listed in the following sub-sections:

A. Baseline CNN Model

B. ResNet50

C. ResNet152

D. VGG16

E. Vision Transformer (ViT)

IV. Experimental setup

In this section, we will outline our experimental settings. We have divided this section into 5 parts i.e., Hardware, Software, Dataset Preparation, Hyperparameters, and Metrics.

A. Hardware Settings: All the experiments were conducted on an ASUS G14 machine that comprised of an AMD-Ryzen 9/4000-series CPU, RTX-2060 MAXQ GPU, and 16GB DDR4 RAM.

B. Software Settings: We used PyTorch (cite) as the primary deep learning package. Furthermore, secondary utility libraries such as Matplotlib, Sklearn, Numpy, Pandas, Seaborn, Tqdm, and vit_pytorch (cite all of em) were also put to use.

C. Data Preparation: We first acquired the dataset from Kaggle (cite). Then we selected 7 relevant classes from it and saved it on our repository. Following this, we divided our dataset of 3,737 image samples into training, validation, and testing sets in the ratio of 80-10-10 respectively. The split resulted in 2989 training images, 374 validation images, and 374 testing images. Furthermore, we extracted the mean and standard deviation of the training set images and normalized the training set, validation set, and the testing set with the obtained values. Finally, we augmented the data by first resizing it to 224 x 224 dimensions, then by adding random rotations, random horizontal and vertical flips, and converting them to tensors for faster computations.

D. General Training Hyperparameters: The performance of any deep learning model depends highly on certain hyperparameters. After careful observations of the performance of the models on various hyperparameters, we obtained a learning rate of 0.001, weight Decay of 0.0001, and a batch size of 8. We trained all our models on 25 epochs. Moreover, to avoid overfitting we deployed two callbacks - ReduceLROnPlateau (patience=5, factor=0.01) and saved the best model monitored on ‘Val_accuracy’.

E. Metrics: We evaluated the performance of the models statistically and visually on accuracy, precision, recall, and f1-scores.

The accuracy refers to the ratio of total correctly classified samples to the total number of samples.

$$Accuracy = \frac{1}{n_{sample}} \sum_{i=0}^{n_{sample}} l(\hat{y_i} = y_i) \tag{1}$$

$$Precision = \frac{tp}{tp + fp} \tag{2}$$

$$Recall = \frac{tp}{tp + fn} \tag{3}$$

$$F_1 = 2 * \frac{PR}{P + R} \tag{4}$$

where $\hat{y_i}$ represents the predicted class and y_i represents the actual class.

Precision refers to the ability of the model to predict number of positive class predictions that actually belong to the positive class and Recall refers to the number of positive class predictions made out of all positive examples in the dataset. Furthermore, f1-score is the harmonic mean of recall and precision with β = 1 giving equal importance to both recall and precision. Equations (2), (3), and (4) illustrates these quantitative metrics.

where the notions of P, R, F_1 are defined by using true positives (tp), true negatives (tn), and false positives (fp) (CITE Zhou 2016)

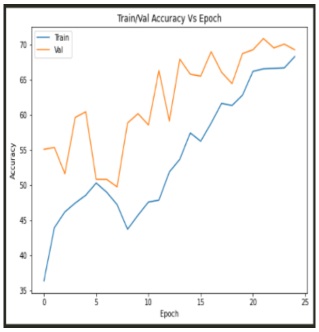
Model Name	Total Parameters	Trainable parameters	Non-trainable parameters	Estimated Size (MB)	Test Accuracy
Baseline	134,467,463	134,467,463	0	1026.64	77.54%
Resnet50	25,613,383	2,105,351	23,508,032	847.96	88.77%
Resnet152	60,249,159	2,105,351	58,143,808	1816.08	72.73%
MobilenetV2	3,145,991	922,119	2,223,872	412.99	80.48%
Vgg16	138,463,047	20,983,81	117,479,232	1103.16	90.91%
ViT	**85,651,975**	**5,383**	**85,646,592**	**2482.32**	**97.86%**

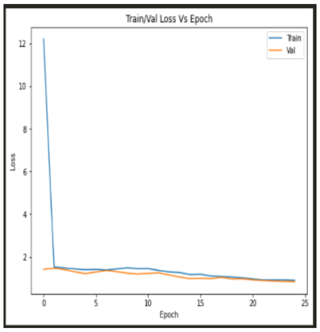
V. RESULTS AND PERFORMANCE ANALYSIS

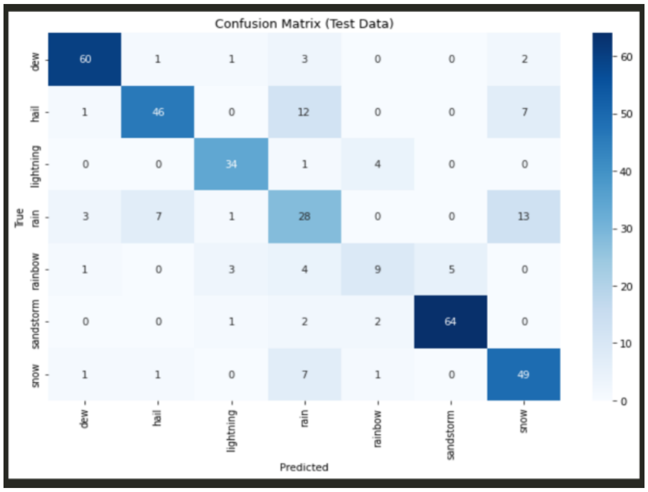
The tests on the unseen portion of the dataset revealed some variability in the performances of the models, which will be elaborated further upon in this section.

A. Model Parameter Comparison: As shown in Table 1, which lists the number of parameters for each model along with the test accuracy achieved, the Vision Transformer's ability to focus on prominent aspects of images through attention gave it an edge over the other models, yielding a test accuracy of 97.86% despite having the least trainable parameters. The VGG16 was the closest contender albeit by a significant margin, showing an impressive 90.91% accuracy even with only 16 layers.

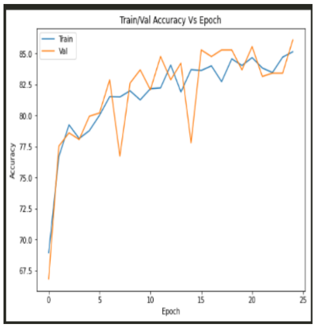
B. Baseline CNN performance: The 'MeteCNN' network initially experienced a significant gap between the training and validation accuracy, which narrowed gradually with the number of epochs as shown in figure (). Although the training and validation losses were consistently low and close to one another as in figure (), the model achieved around 70% validation accuracy and a 77.54% test accuracy, signaling possible overfitting. Moreover, the loss graph shows it to be suboptimal within the 1.0 to 2.0 range. Out of the classes, 'snow', 'rain', and 'rainbow' were misclassified the most due to a relatively low number of training samples for those classes.

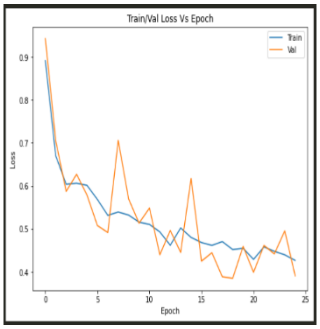


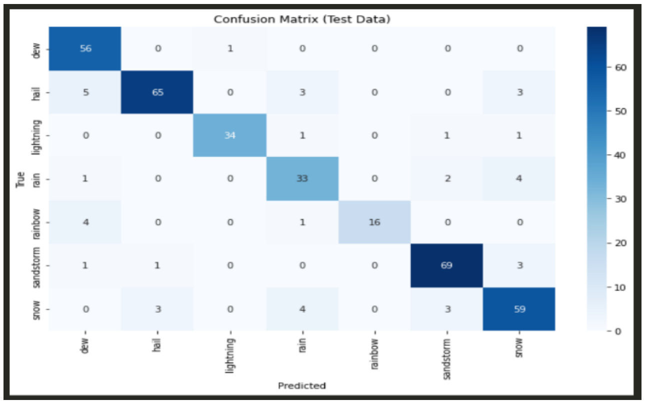




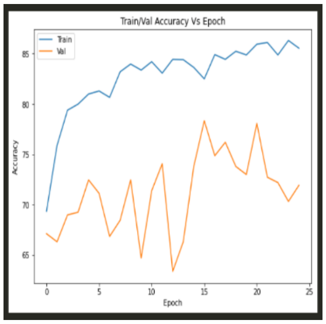
C. ResNet50: Possessing a large number of convolution layers, the ResNet50 fluctuated within the 75-86% training accuracy range over the 25 epochs, with the validation accuracy also being consistently close in value. The loss yielded was also lower than the MeteCNN, with the validation and training losses between the 0.4 to 0.65 range. The confusion matrix indicates an overall better precision for each class, especially for the uncommon 'rainbow' class which had all its samples classified correctly. The classes 'snow', 'dew', and 'rain' were the most misclassified, although the model achieved better recognition performance than the MeteCNN model.

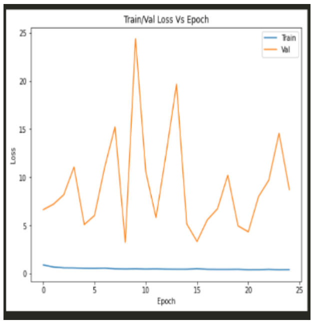


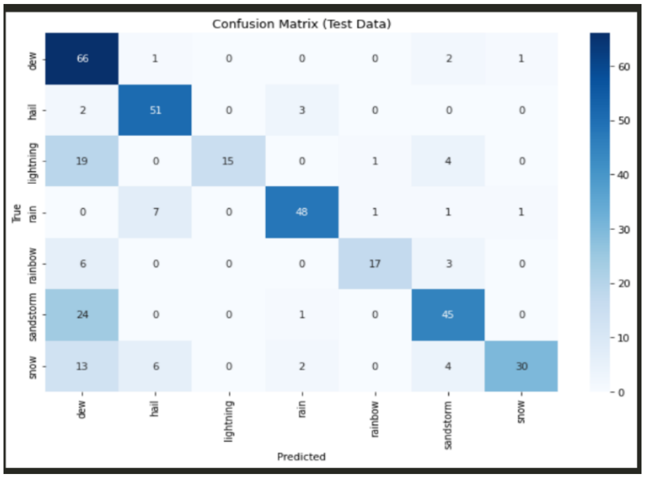




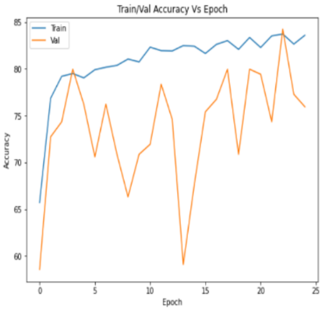
D. ResNet152: Despite consisting of a larger number of layers, the ResNet152 did not perform as well as its 50 layer counterpart. Drastic gaps between the training and validation accuracy as depicted in figure (), along with high validation loss as in figure () show signs of overfitting. Being the poorest performer among the models trained for 25 epochs, the ResNet152 achieved relatively low precision for each class, even misclassifying many of the samples from the common 'dew' class.

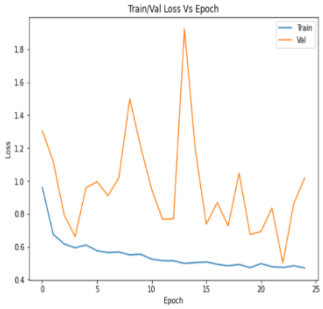


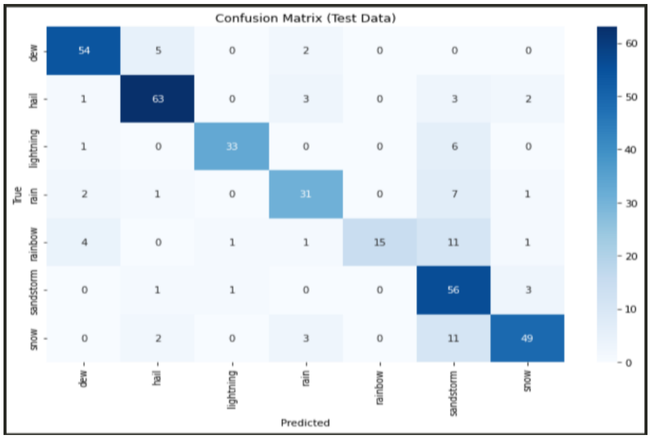




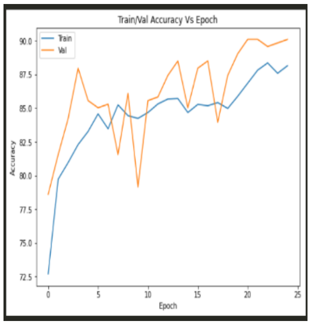
E. MobileNetV2: While performing better than the ResNet152, the MobileNet suffered huge drops in validation accuracy while having a consistently high training accuracy as depicted in figure (). The loss graph in figure () indicates overfitting through the noticeable gap between the decreasing training loss and the disorderly fluctuation of the validation loss. The confusion matrix in figure () indicates that the 'sandstorm' class yielded the lowest precision rate, followed by 'snow' and 'hail'. However, the 'lightning' and 'rainbow' classes experienced the highest precision rates.

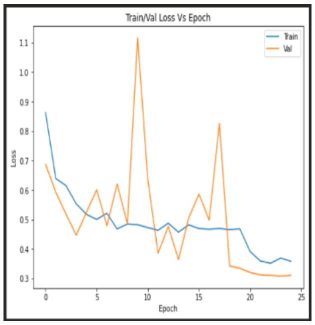


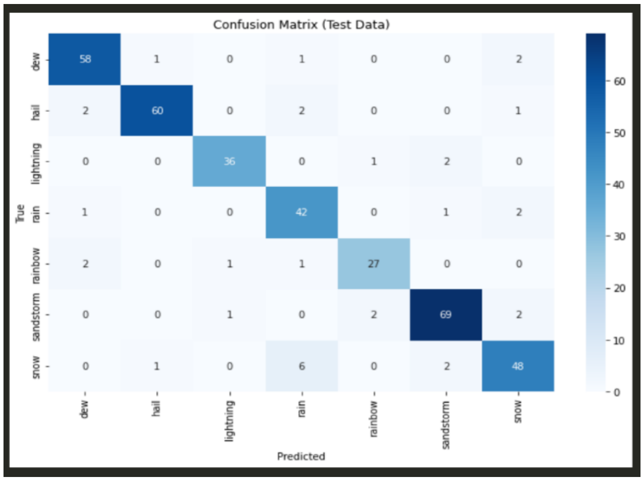




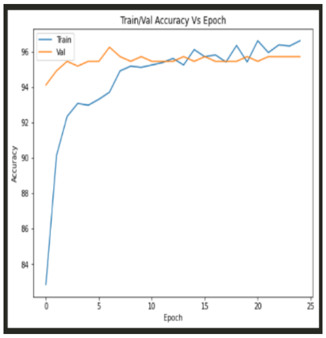
F. VGG16: The simple yet concise structure of the VGG16 was enough to make it the second best performing model in the experiment. Validation accuracies were at times higher than the training accuracy (possibly due to the 80-10-10 split), hovering between 85 to 90% as shown in figure (). Although the loss graph shows spikes in validation loss, it still mostly resides within the 0.3 to 1.0 range, indicating lower loss on average. Th confusion matrix depicted in figure () shows an almost perfect blue diagonal across the classes, implying high precision rates.

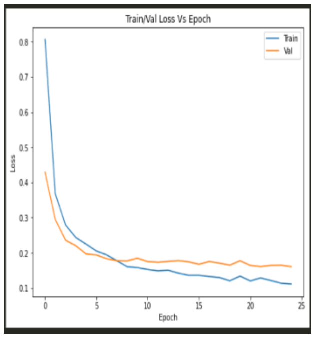




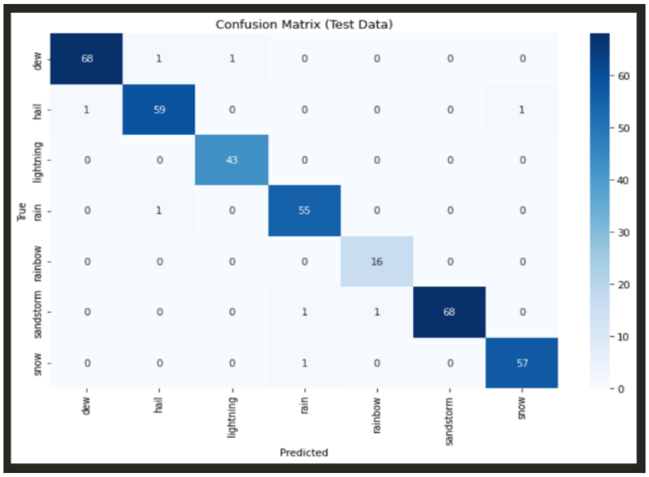


G. Vision Transformer (ViT): Using attention based layers to prioritize important image features, the Vision Transformer model achieved excellent consistency between the training and validation accuracies across the 25 epochs as shown in figure (). The model also experienced a steady, asymptotic loss curve for both training and validation as depicted in the graph in figure (). Extremely high precision rates for all classes were achieved, with a 100% for the ‘sandstorm’ class, and 94-98% rates for the other classes. The ‘rainbow’ class had the lowest precision of around 94.12%, lower than that of the MobileNetV2. Overall, the ViT attained the best performance among the experimented models as reflected by the confusion matrix in figure ().





H. Error Analysis: Despite achieving the best average performance, the Vision Transformer model was possibly confused between the ‘snow’ and ‘hail’ classes, leading to imperfect precision scores. This may be due to the tiny intricacies and details that differentiate the two weather phenomena. We believe the addition of more attention layers and prolonged training by increasing the number of epochs would help in familiarizing the model with the traits of each class.



V. CONCLUSION

ACKNOWLEDGEMENTS

REFERENCES