King Fahd University of Petroleum and Minerals
Information and Computer Science Department

ICS 471: Artificial Neural Networks and Deep Learning

Term 211

Middle Eastern Weather Classification Using Deep Learning

Name	ID
Farhan M. Abdul Qadir	**201771950**
Amaan Izhar	**201781130**
Abdul Jawad Mohammed	**201744310**

I. Introduction

Throughout human history, the weather has played an integral role in shaping our lifestyles across different parts of the world, both as a boon and a curse. Rainfall is one such example, giving rise to numerous crops through the invigoration of the land, yet in excess causing floods, endangering traffic safety, and disrupting our day-to-day lives. Having an educated foresight into the weather in the near future is a great asset to help us prepare and plan out one's schedule accordingly. Moreover, it could aid in deciding the suitability of keeping public facilities like schools, roads, and factories open. With the occurrence of unpredictable climate patterns and an increase in human activity around the globe, the significance of weather detection has become very apparent for environmental safety, observation of climate change, and forecasts.

II. Problem Description

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 could bring abrupt changes to school and employee schedules but could also pose a threat to traffic safety and infrastructure due to lack of coordination and sophisticated sewage systems.

To improve weather recognition technology for use in public, we are to use Deep Learning techniques spanning computer vision transformers, transfer learning, and a baseline Convoluted Neural network, to train several models on a weather image dataset containing classes mostly relevant to Middle Eastern weather, before recording, analyzing, and comparing the results with the custom 'MeteCNN' model published by Xiao H. et al. (2021) during their research on Weather image recognition.

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.

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

IV. Experimental Setup

Our experimental setup included the following hardware/software/libraries:

GPU :- RTX 2060 MAXQ

Core Software :- PyTorch

Supplementary libraries :- Pandas, NumPy, Matplotlib, Seaborn, Sklearn

Hyperparameters :-

- Learning Rate: 0.001
- Weight Decay: 0.0001
- Batch Size: 8
- Epochs: 25
- Callbacks: ReduceLROnPlateau (patience=5, factor=0.01), Saved best model (monitored on val_accuracy).

The dataset was first augmented. Random rotations, horizontal & vertical flips, and resizing were applied to each image, before being normalized with their mean and standard deviation extracted. Furthermore, 80% of the images were used for training, 10% for validation, and 10% for testing. The split resulted in 2989 training images, 374 validation images, and 374 testing images. It took more than 8 hours to fine-tune and train all the models.

V. Deep Learning Models

1. CNNs

1.1 Normalized CNN (Baseline model)

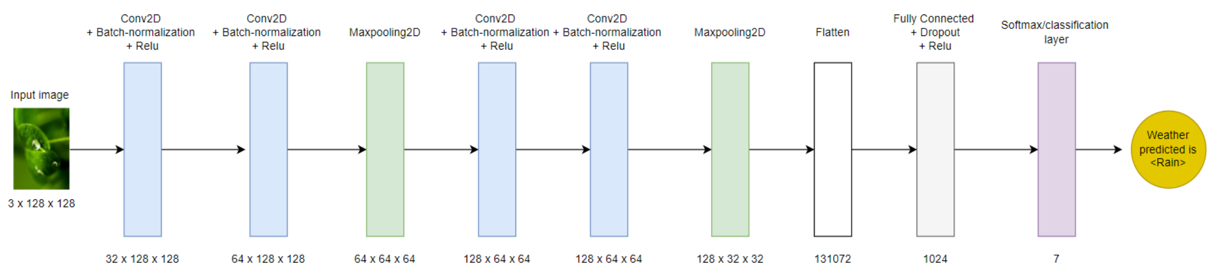


Fig 2. Baseline CNN Model using Conv Blocks

This model serves as a baseline for benchmarking other models. The primary feature of this model is that batch-normalization is used after every convolution layer, making parameters non-sensitive to changes and keeping the coherency of the weights/biases.

2. Transfer Learning

Transfer learning is a technique that involves the training of pre-trained models in combination with user-specified fine-tuned layers. The main advantage sweeps in when there is a requirement of less trainable parameters. Hence, we harnessed the power of SOTA models and utilized 4 such models. Their architecture is explained by the figures below.

2.1 Resnet50

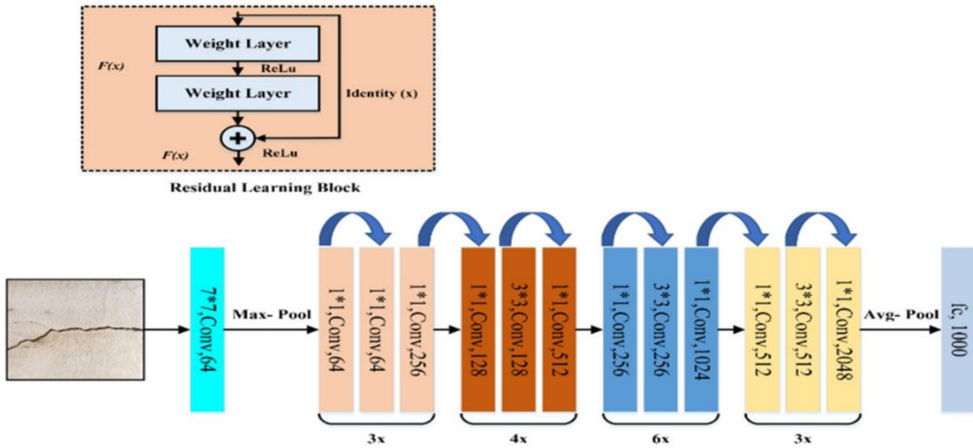


Fig 3. Residual networks with a depth of 50 layers.

2.2 Resnet152

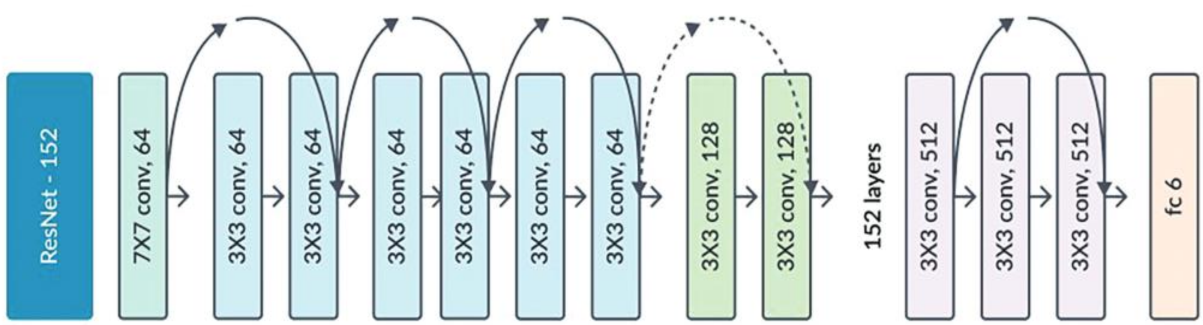


Fig 4. Residual networks with a depth of 152 layers.

2.3 MobileNet V2

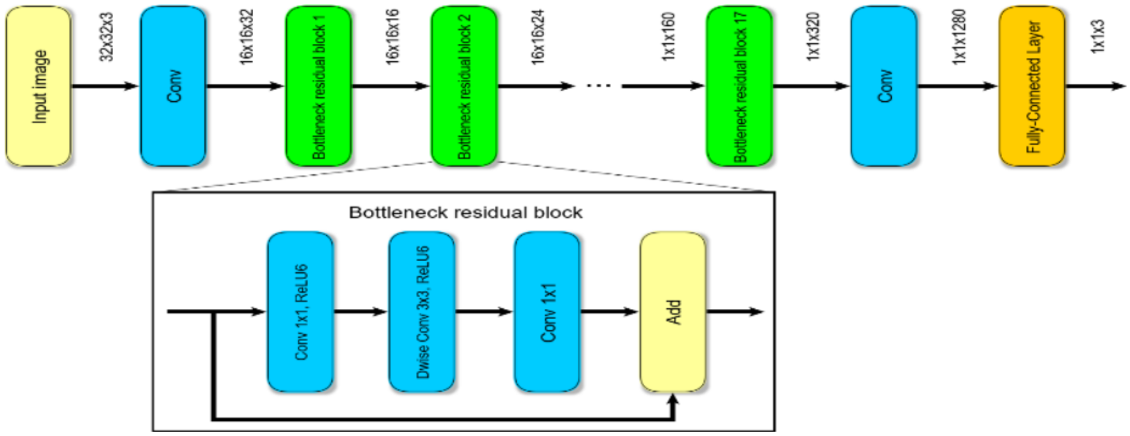


Fig 5. MobileNetV2 models use pointwise and depthwise convolution techniques

2.4 Vgg16

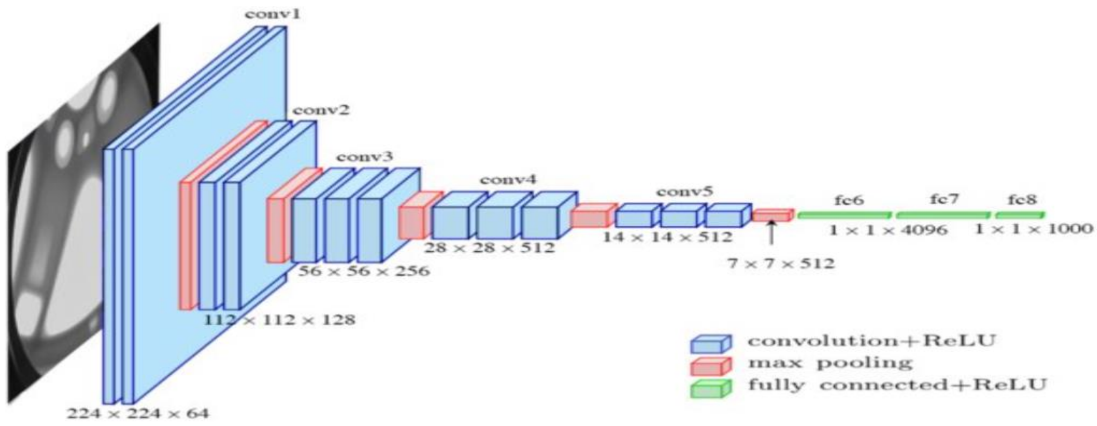


Fig 6. The VGG16 model. Multiple convolutions stacked architecture.

3. Transformers (Vision)

ViT or Vision transformer is a recent addition to the SOTA computer vision models. The primary enticing point of ViT is that it applies the concept of attention and transformers on the images. It divides the image into patches of pixels and pays attention to those patches. After training, the model learns which parts of the image to pay attention to. Since, it was released two months ago, we thought to try it and fine tune it towards our needs. The architecture is shown below.

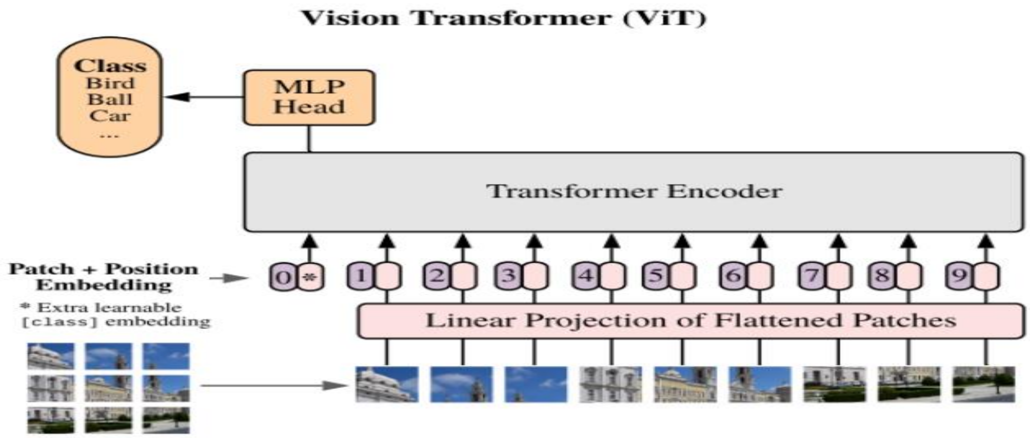


Fig 7. An application of a vision Transformer model on images.

VI. Result Analysis & Discussion

In this section, we will summarize the results followed by some discussions.

6.1 Plots – Loss and Accuracy per epoch

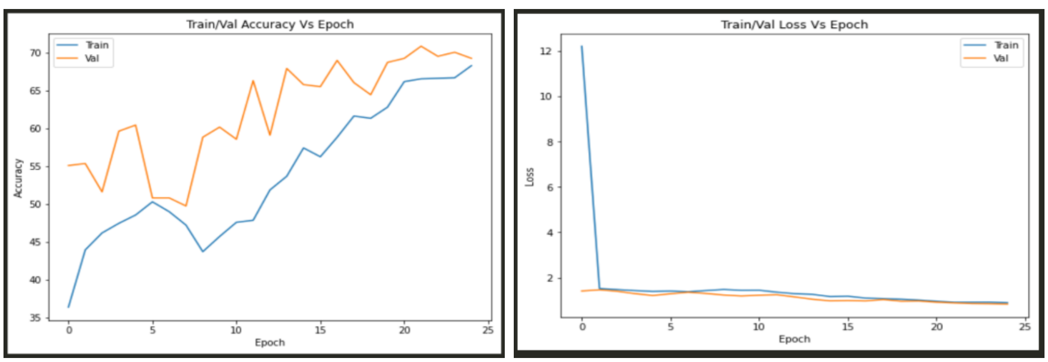


Fig 8. Normalized CNN performance graphs

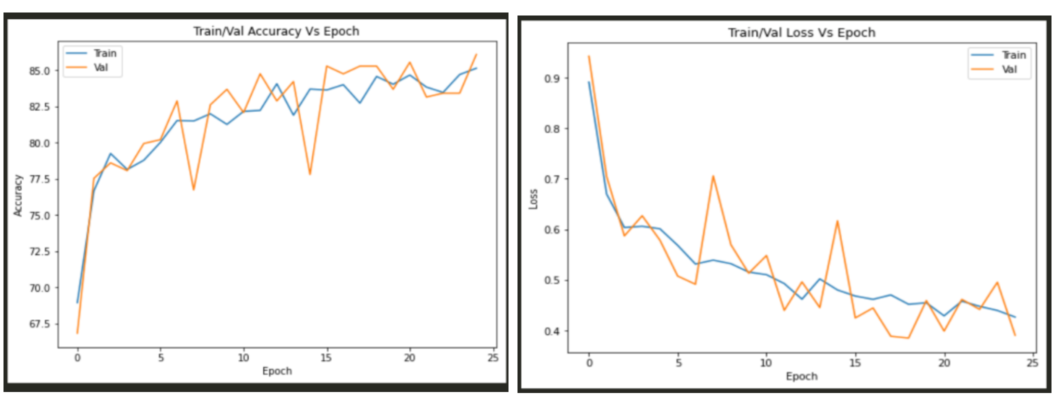


Fig 9. ResNet50 performance graphs

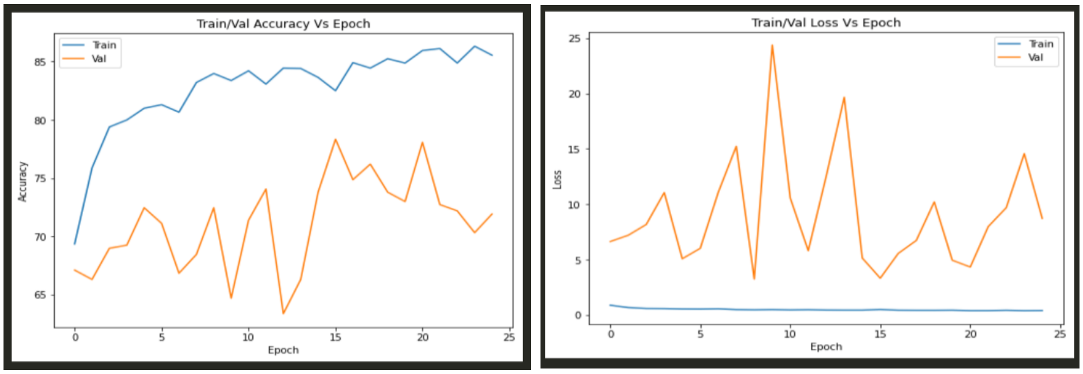


Fig 10. ResNet152 performance graphs

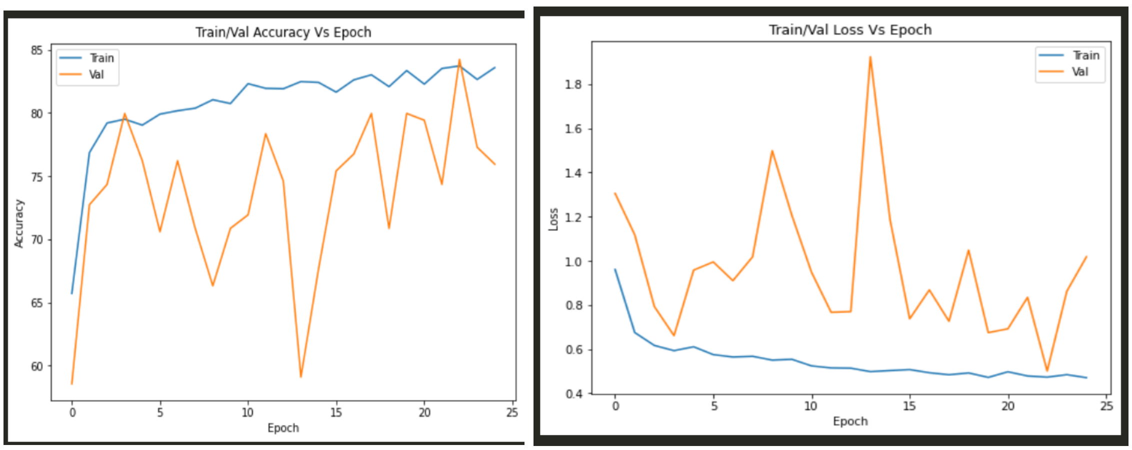


Fig 11. MobileNetV2 performance graphs

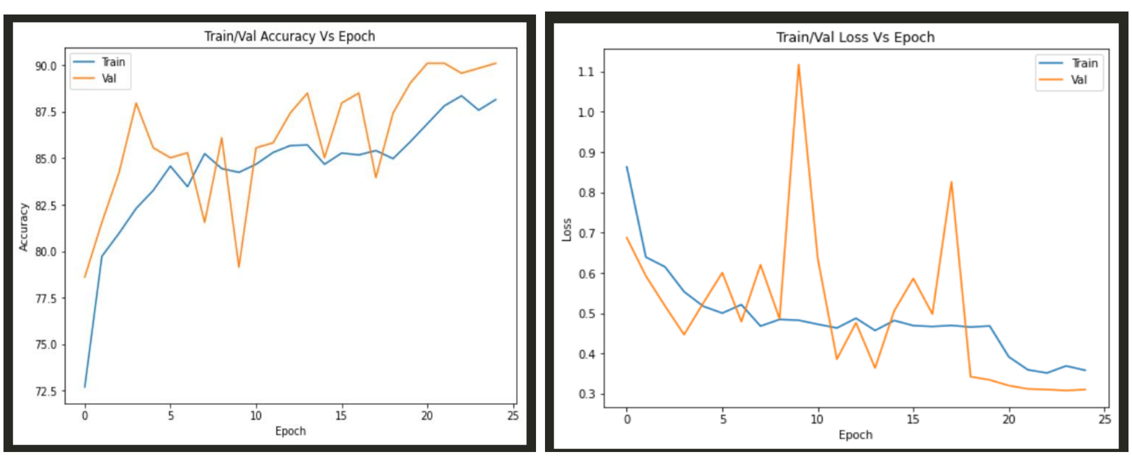


Fig 12. VGG16 performance graphs

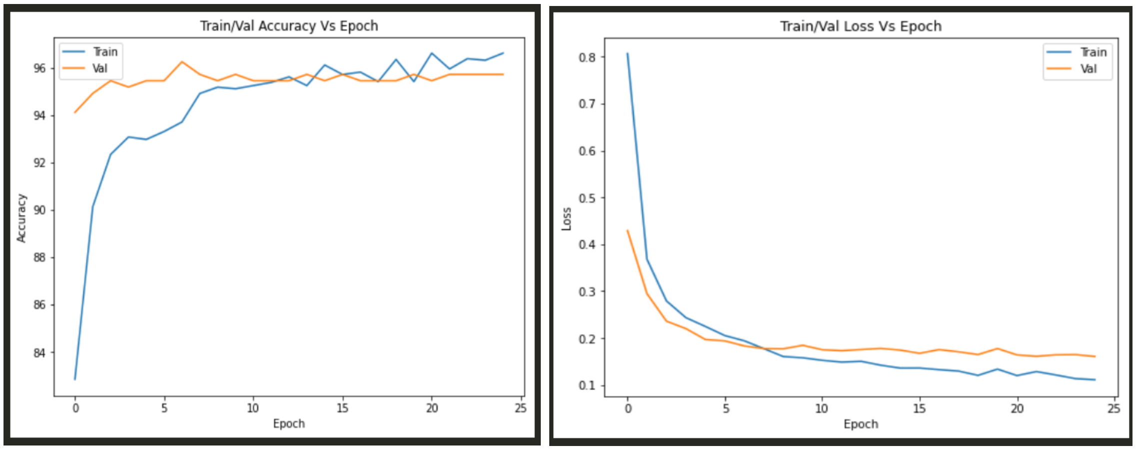


Fig 13. Vision Transformer (ViT) performance graphs

6.2 Confusion matrices

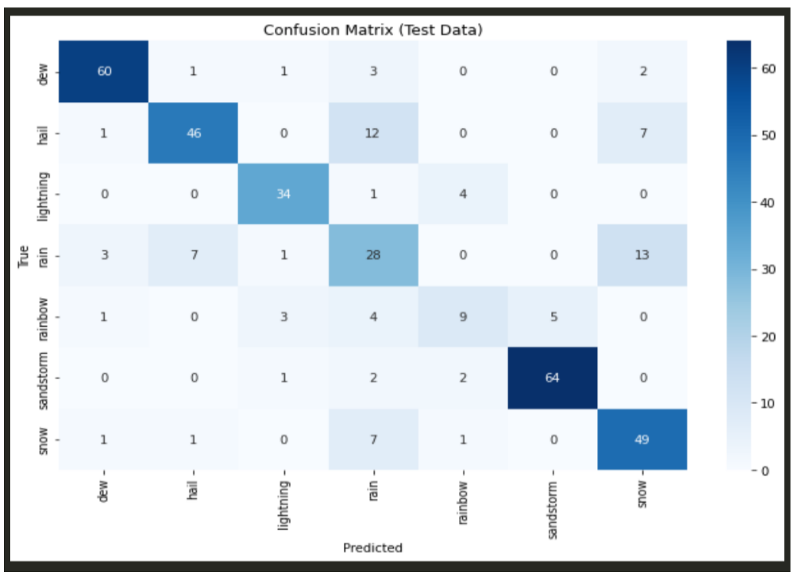


Fig 14. Normalized CNN Confusion Matrix

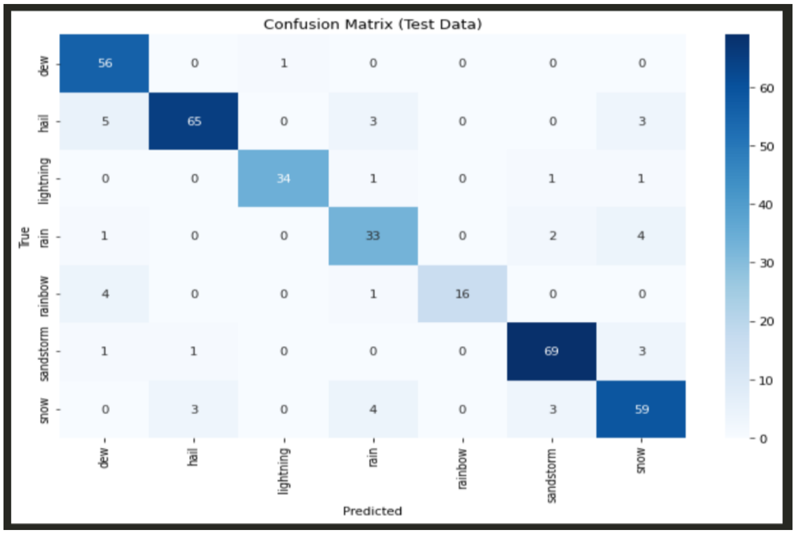


Fig 15. ResNet50 performance graphs

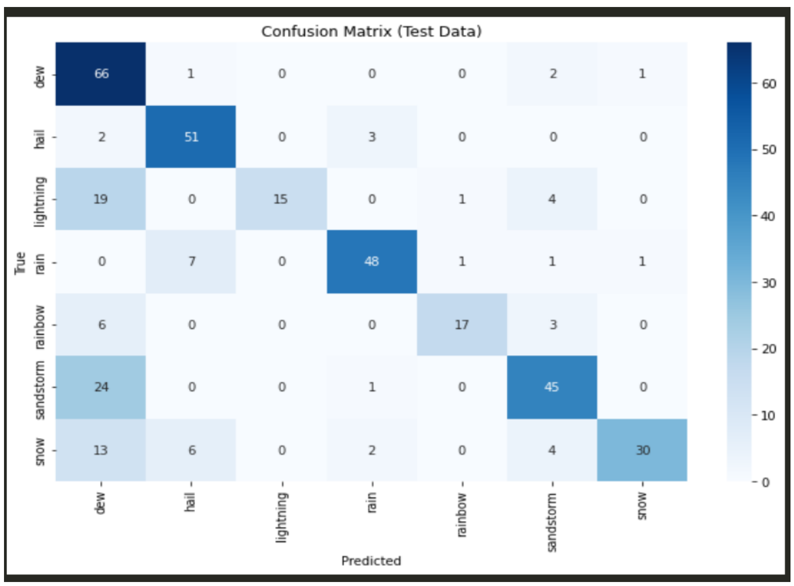


Fig 16. ResNet152 Confusion Matrix

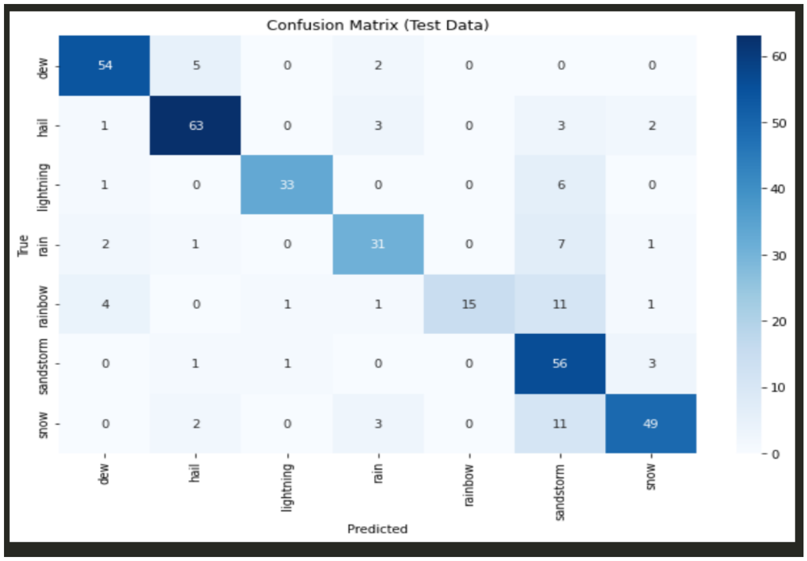


Fig 17. MobileNetV2 Confusion Matrix

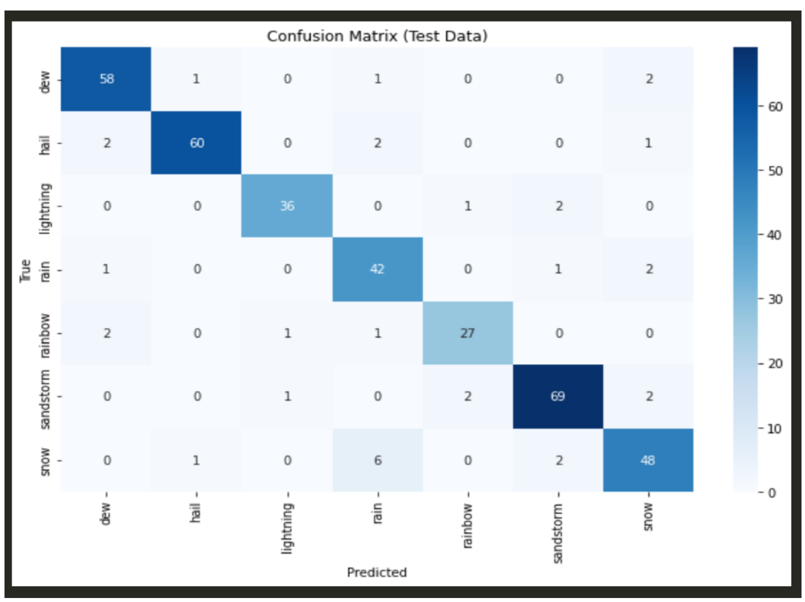


Fig 18. VGG16 Confusion Matrix

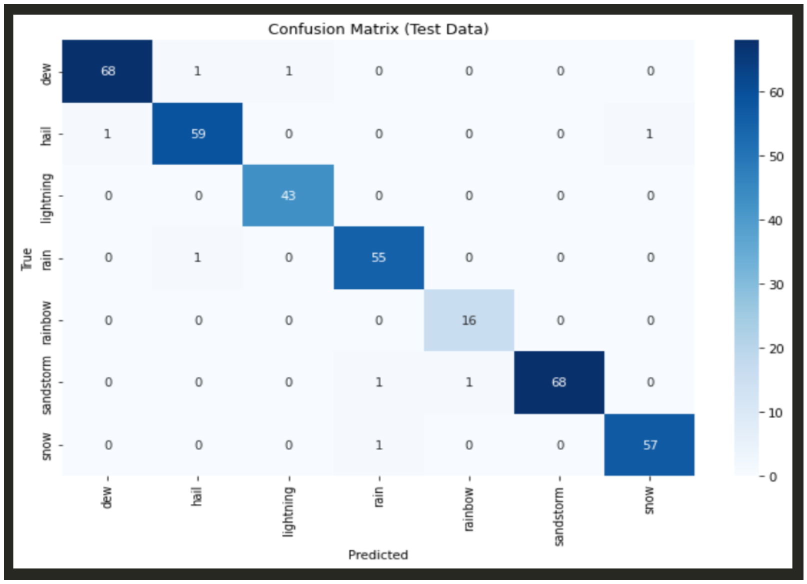


Fig 19. Vision Transformer (ViT) Confusion Matrix

6.3 Classification report

Classification report:

	precision	recall	f1-score	support
dew	0.909091	0.895522	0.902256	67.000000
hail	0.836364	0.696970	0.760331	66.000000
lightning	0.850000	0.871795	0.860759	39.000000
rain	0.491228	0.538462	0.513761	52.000000
rainbow	0.562500	0.409091	0.473684	22.000000
sandstorm	0.927536	0.927536	0.927536	69.000000
snow	0.690141	0.830508	0.753846	59.000000
accuracy	0.775401	0.775401	0.775401	0.775401
macro avg	0.752409	0.738555	0.741739	374.000000
weighted avg	0.780471	0.775401	0.774909	374.000000

Fig 20. Normalized CNN Classification Report

Classification report:

	precision	recall	f1-score	support
dew	0.835821	0.982456	0.903226	57.000000
hail	0.942029	0.855263	0.896552	76.000000
lightning	0.971429	0.918919	0.944444	37.000000
rain	0.785714	0.825000	0.804878	40.000000
rainbow	1.000000	0.761905	0.864865	21.000000
sandstorm	0.920000	0.932432	0.926174	74.000000
snow	0.842857	0.855072	0.848921	69.000000
accuracy	0.887701	0.887701	0.887701	0.887701
macro avg	0.899693	0.875864	0.884151	374.000000
weighted avg	0.892633	0.887701	0.887797	374.000000

Fig 21. ResNet50 Classification Report

Classification report:

	precision	recall	f1-score	support
dew	0.507692	0.942857	0.660000	70.000000
hail	0.784615	0.910714	0.842975	56.000000
lightning	1.000000	0.384615	0.555556	39.000000
rain	0.888889	0.827586	0.857143	58.000000
rainbow	0.894737	0.653846	0.755556	26.000000
sandstorm	0.762712	0.642857	0.697674	70.000000
snow	0.937500	0.545455	0.689655	55.000000
accuracy	0.727273	0.727273	0.727273	0.727273
macro avg	0.825164	0.701133	0.722651	374.000000
weighted avg	0.797454	0.727273	0.725134	374.000000

Fig 22. ResNet152 Classification Report

Classification report:

	precision	recall	f1-score	support
dew	0.870968	0.885246	0.878049	61.000000
hail	0.875000	0.875000	0.875000	72.000000
lightning	0.942857	0.825000	0.880000	40.000000
rain	0.775000	0.738095	0.756098	42.000000
rainbow	1.000000	0.454545	0.625000	33.000000
sandstorm	0.595745	0.918033	0.722581	61.000000
snow	0.875000	0.753846	0.809917	65.000000
accuracy	0.804813	0.804813	0.804813	0.804813
macro avg	0.847796	0.778538	0.792378	374.000000
weighted avg	0.835852	0.804813	0.804450	374.000000

Fig 23. MobileNetV2 Classification Report

Classification report:

	precision	recall	f1-score	support
dew	0.920635	0.935484	0.928000	62.000000
hail	0.967742	0.923077	0.944882	65.000000
lightning	0.947368	0.923077	0.935065	39.000000
rain	0.807692	0.913043	0.857143	46.000000
rainbow	0.900000	0.870968	0.885246	31.000000
sandstorm	0.932432	0.932432	0.932432	74.000000
snow	0.872727	0.842105	0.857143	57.000000
accuracy	0.909091	0.909091	0.909091	0.909091
macro avg	0.906942	0.905741	0.905702	374.000000
weighted avg	0.911041	0.909091	0.909490	374.000000

Fig 24. VGG16 Classification Report

Classification report:

	precision	recall	f1-score	support
dew	0.985507	0.971429	0.978417	70.00000
hail	0.967213	0.967213	0.967213	61.00000
lightning	0.977273	1.000000	0.988506	43.00000
rain	0.964912	0.982143	0.973451	56.00000
rainbow	0.941176	1.000000	0.969697	16.00000
sandstorm	1.000000	0.971429	0.985507	70.00000
snow	0.982759	0.982759	0.982759	58.00000
accuracy	0.978610	0.978610	0.978610	0.97861
macro avg	0.974120	0.982139	0.977936	374.00000
weighted avg	0.978883	0.978610	0.978633	374.00000

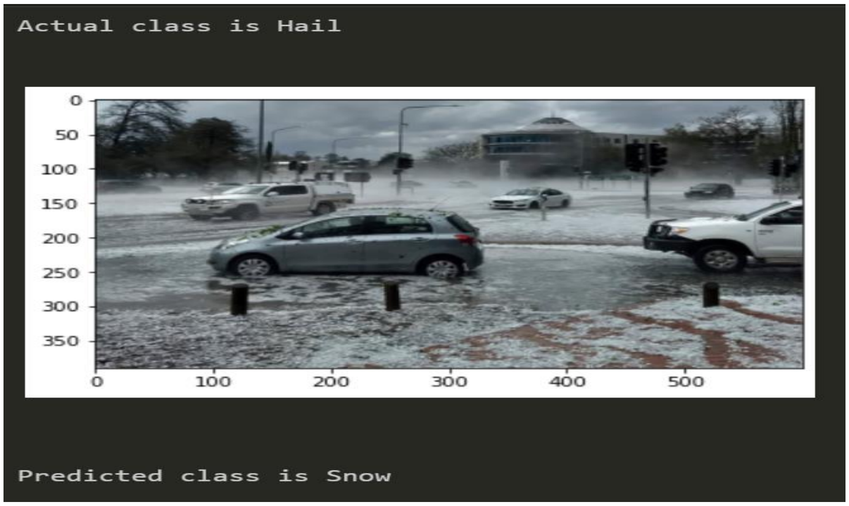
Fig 25. Vision Transformer (ViT) Classification Report

6.4 Model params, size, and accuracy comparison

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%**

6.5 Error sources

Our best model i.e. ViT got confused between snow and hail. The reason being that there are very tiny intricacies and details that differentiates the two-weather phenomena. We think that training the model for more epochs and applying more attention layers could mitigate this problem. The following image illustrates this error:



7. Conclusion

Despite having the lowest number of trainable parameters, the Computer Vision Transformer (ViT) attained the best overall performance for training, validation, and test sets, outpacing the next highest model (VGG16) by about 7.8%. Its incorporation of attention techniques for focusing on the most important patches of each image proved to be extremely advantageous, efficiently extracting significant features for highly precise classifications. Its strong precision is evident in Fig.19, displaying the Vision Transformer's confusion matrix results.

REFERENCES

1. Ali, L.; Alnajjar, F.; Jassmi, H.A.; Gochoo, M.; Khan, W.; Serhani, M.A. Performance Evaluation of Deep CNN-Based CrackDetection and Localization Techniques for Concrete Structures.Sensors2021,21, 1688
(PDF) An Overview of Challenges Associated with Automatic Detection of Concrete Cracks in the Presence of Shadows. Available from: https://www.researchgate.net/publication/356721805_An_Overview_of_Challenges_Associated_with_Automatic_Detection_of_Concrete_Cracks_in_the_Presence_of_Shadows [accessed Dec 19 2021].

2. Papers with code - vision transformer explained. Explained | Papers With Code. (n.d.). Retrieved December 19, 2021, from https://paperswithcode.com/method/vision-transformer

3. *Pytorch*. PyTorch. (n.d.). Retrieved December 19, 2021, from https://pytorch.org/

4. Ferguson, M., R. Ak, Y.-T.T. Lee, and K.H. Law. Automatic localization of casting defects with convolutional neural networks. 2005. IEEE
(PDF) Integration of Building Information Modeling and machine learning for railway defect localization. Available from: https://www.researchgate.net/publication/357010003_Integration_of_Building_Information_Modeling_and_machine_learning_for_railway_defect_localization [accessed Dec 19 2021]

5. Kramberger, T., Cesar, I., & Kovačević, R. (2020). *(PDF) automobile classification using transfer learning on ...* Retrieved December 19, 2021, from https://www.researchgate.net/publication/343615852_Automobile_Classification_Using_Transfer_Learning_on_ResNet_Neural_Network_Architecture

6. Seidaliyeva, U.; Akhmetov, D.; Ilipbayeva, L.; Matson, E.T. Real-Time and Accurate Drone Detection in a Video with a StaticBackground.Sensors2020,20, 3856
(PDF) Advanced Computational Intelligence for Object Detection, Feature Extraction and Recognition in Smart Sensor Environments. Available from: https://www.researchgate.net/publication/347942689_Advanced_Computational_Intelligence_for_Object_Detection_Feature_Extraction_and_Recognition_in_Smart_Sensor_Environments [accessed Dec 19 2021]