Weather Classification Report

Dylan McIntosh

2/24/24

This project utilizes transfer learning from ResNet50 on a dataset containing images of different weather conditions. The dataset used can be downloaded here https://www.kaggle.com/datasets/jehanbhathena/weather-dataset?resource=download

Table of contents

1 Problem Statement

Tracking the weather and natural disasters is a difficult task. It could help government agencies respond to extreme weather events and assist in weather forecasting. The problem can begin to be tackled with utilizing image recognition to classify images of different weather conditions. Weather being so varied and the many different classes of weather create a complex problem that traditional machine learning won't be able to solve, so a deep learning solution is needed. This is proven in the data section of the report.

2 Data

The dataset contains 6862 images of 11 different classes. The images all come in varying sizes, so some image resizing is needed. A very small number of images are labeled as .jpg files, however they are GIF files.

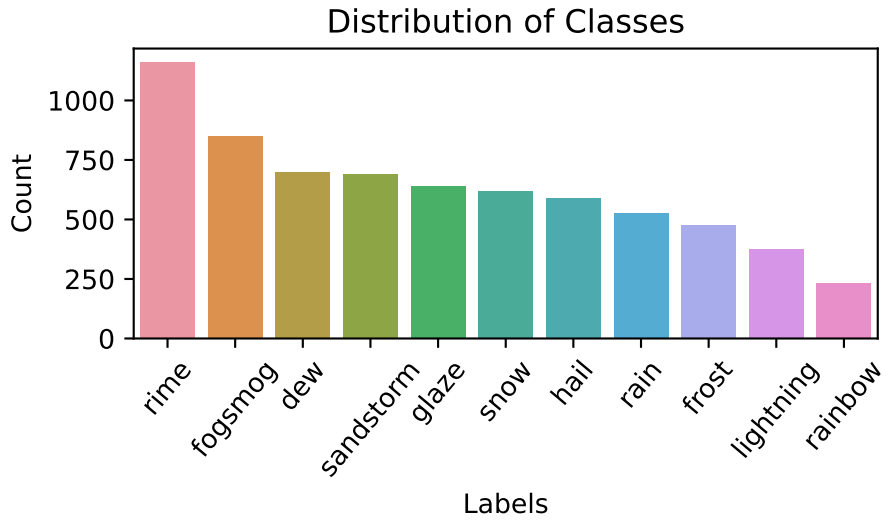
The classes are as follows: ['rain' 'glaze' 'rime' 'rainbow' 'fogsmog' 'frost' 'sandstorm' 'hail' 'snow' 'dew' 'lightning']

Examples of these images can be seen in the appendix.

Some of the classes aren't easily distinguishable from one another with the human eye and could be up to interpretation. For example, the rime, snow, and frost classes are similar. Distinguishing these classes may cause some trouble for the model and evaluation should ensure these are handled properly.

The different colors appear to hold extremely important information. An example is in distinguishing between a sandstorm and a fogsmog label.

The distribution between the classes can be seen here:



Most of the images are rime, with there being a relatively small amount of rainbow images. Ideally there would be more data for each class since under 800 for most classes isn't an effective amount especially before the data split is done.

An exploratory random forest model was used to determine necessity of a deep learning solution.

`Classification Report:`

Table 1

	precision	recall	f1-score	support
dew	0.64	0.75	0.69	140
fogsmog	0.64	0.86	0.73	170
frost	0.44	0.20	0.28	95
glaze	0.50	0.41	0.45	128
hail	0.59	0.61	0.60	118
lightning	0.83	0.82	0.82	76
rain	0.54	0.43	0.48	105
rainbow	0.71	0.22	0.33	46
rime	0.60	0.76	0.67	232
sandstorm	0.76	0.65	0.70	139
snow	0.56	0.54	0.55	124
accuracy	0.62	0.62	0.62	0
macro avg	0.62	0.57	0.57	1373
weighted avg	0.61	0.62	0.60	1373

These above results are not satisfactory for solving this problem. This indicates that a deep learning solution is necessary since the complexity of the problem cannot be captured by traditional machine learning models. It is also important to note that another random forest model was tested on this dataset but using grayscale images. The performance of this grayscale model was much worse, indicating that the additional channels assist greatly in classifying the images.

3 Methods (Preprocessing)

The data was split into train, validation, and test sets with a split of 70/10/20. Stratified splitting was used to ensure stable distributions of classes amongst all three datasets. Generators were created with several data augmentation methods to improve generalizability of model. Rotation, zoom, dimension shifts, shearing, and horizontal flipping were used as augmentation. All images were resized to 224x224 since that is the input size ResNet50 was trained with along with the 3 RGB color channels since those are very important in classifying.

The few cases of GIF files hidden in the data were converted into the first frame of the animation as a .jpg file.

4 Methods (Model)

Utilized a two step training approach with the first phase being feature extraction, and the second phase being fine tuning.

4.1 Feature Extraction Model

The ResNet50 pre-trained model was loaded from tensorflow libraries with the weights trained from the imagenet dataset. The top layer was replaced with the new architecture. The final layer of ResNet50 leads into a global average pooling layer, then to two dense layers of 100 neurons with relu activation. There are 50% dropout layers applied after each dense layer for regularization. The final output layer is dense with 11 nodes and softmax activation.

The new model is then compiled with categorical crossentropy as the loss, the Adam optimizer, and a default learning rate of 0.001. Early stopping with a patience of 5 is included as well as a model checkpoint to save the best model during training, both with respect to validation loss.

The model was then set to train for 100 epochs with all original pre-trained layers frozen.

4.2 Fine Tuned Model

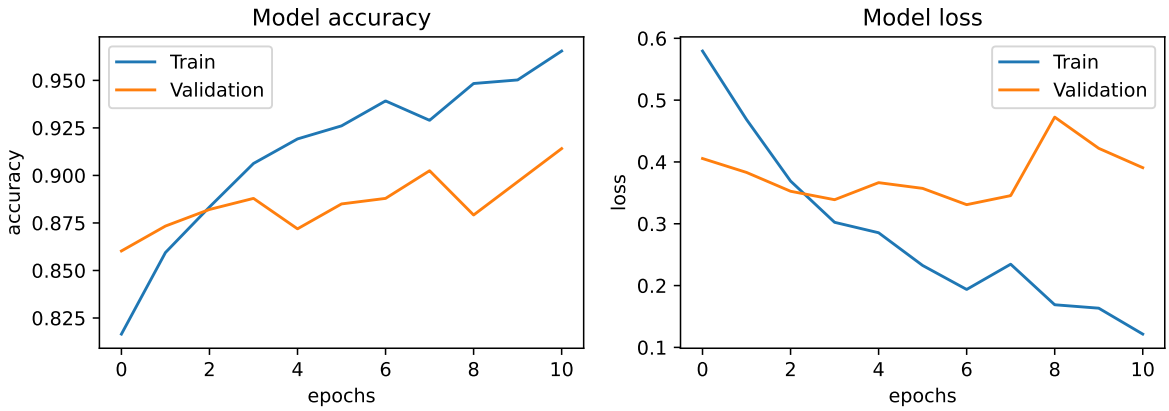
Now that the new layers have been trained, the fine tuning of the pre-trained layers can begin.

The checkpoint model with the best validation loss was loaded in to be altered. All top 50 layers of the model, except for batch normalization layers, were unfrozen to be fine tuned. The batch normalization layers were kept frozen so that the information about distributions learned in the original model isn't lost.

The model was compiled with the same loss and optimizer, except for a new smaller learning rate of 0.0001 to preserve pre-trained weights and encourage convergence. The same callbacks were used to save the best checkpoint model.

The model was then sent to train for 100 epochs.

5 Results

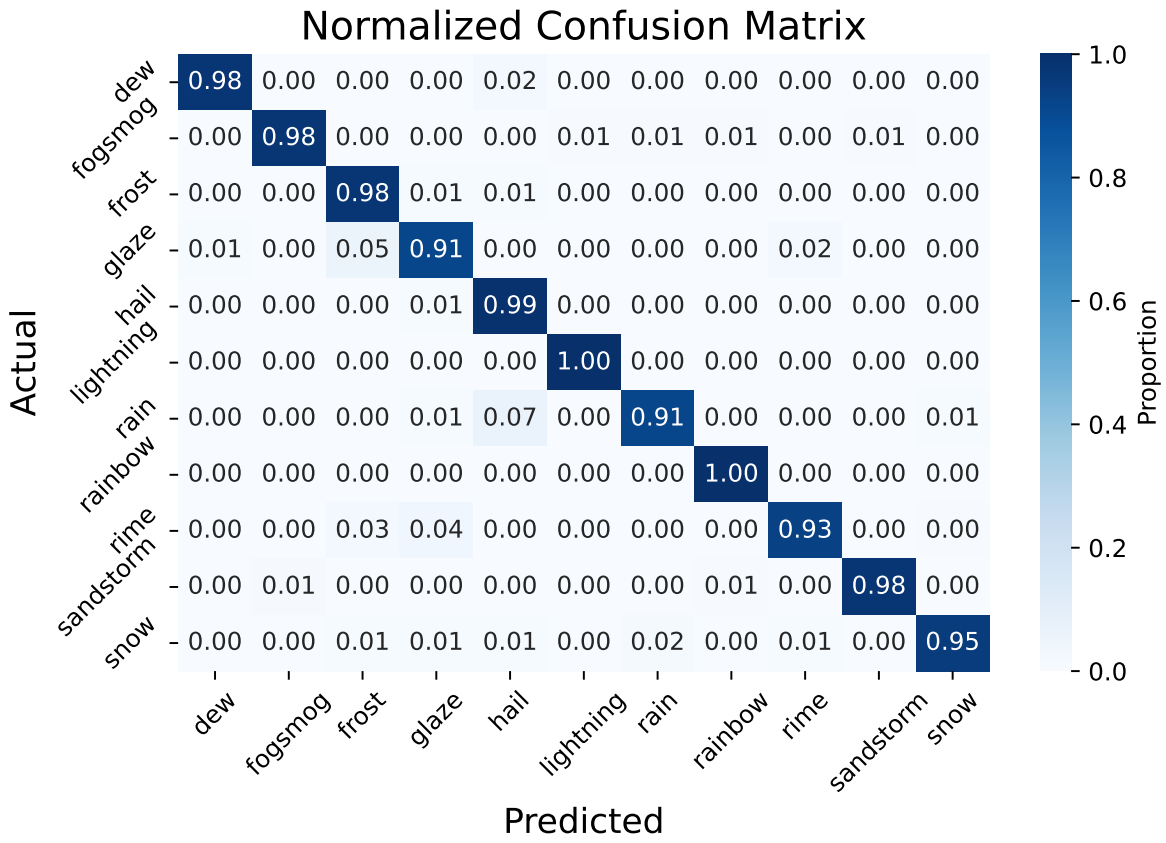


The history of the training above shows that the loss slowly adjusts as the model fine tunes, and the best validation loss was at epoch 6. This means that it is the model that will be used for the rest of the report. It is important to note that the reason the validation set appears to perform better is because there is no dropout being applied during inference, so the model performs much better.

`Classification Report:`

Table 2

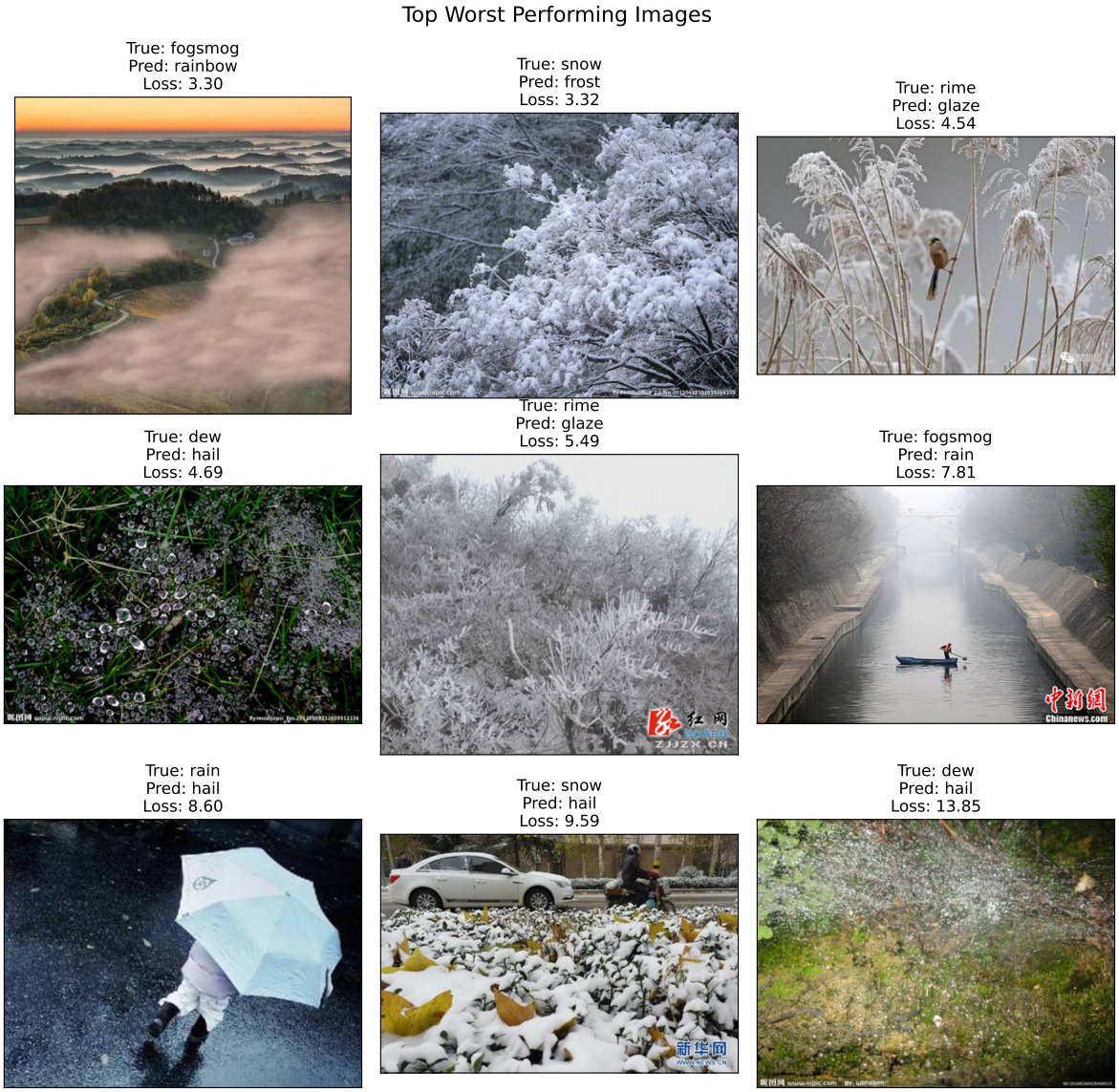
	precision	recall	f1-score	support
dew	0.99	0.98	0.99	140
fogsmog	0.99	0.98	0.98	170
frost	0.87	0.98	0.92	95
glaze	0.90	0.91	0.91	128
hail	0.91	0.99	0.95	118
lightning	0.99	1.00	0.99	76
rain	0.97	0.91	0.94	105
rainbow	0.96	1.00	0.98	46
rime	0.98	0.93	0.96	232
sandstorm	0.99	0.98	0.99	139
snow	0.98	0.95	0.97	124
accuracy	0.96	0.96	0.96	0
macro avg	0.96	0.96	0.96	1373
weighted avg	0.96	0.96	0.96	1373



This model performs significantly better than the random forest model. There is struggle with snow/cold weather conditions, with frost and glaze classes having the worst performance. The confusion matrix shows that the main issues to resolve are the model predicting frost when it should be glaze and predicting hail when it should be rain. An accuracy of 96% along with around that same percentage for macro and weighted averages for all of precision, recall, and f1-score is good performance.

The main takeaway is that the model falters with frost vs glaze and hail vs rain, however great performance otherwise with the confusion matrix focused along the diagonal.

```
<Figure size 2100x1200 with 0 Axes>
```



The above images show how difficult it is for the model to distinguish between the snow/cold classes, whereas the data is difficult for even a human to label.

6 Discussion

With the model's main error being predicting snow/cold conditions incorrectly, this could lead to snowstorms going undetected decently often in the real world. Even though the precision and recall values aren't awful, a value as low as 0.91 likely wouldn't suffice for deployment. Issues like this when

deployed will become emphasized since there would be many inferences happening, and more are probably going to be incorrect inferences than ideal.

Including multiple channels proved to be a great performance enhancer since color differences in weather conditions is extremely relevant. Also adding dropout layers led to a much better performance on the validation set during training

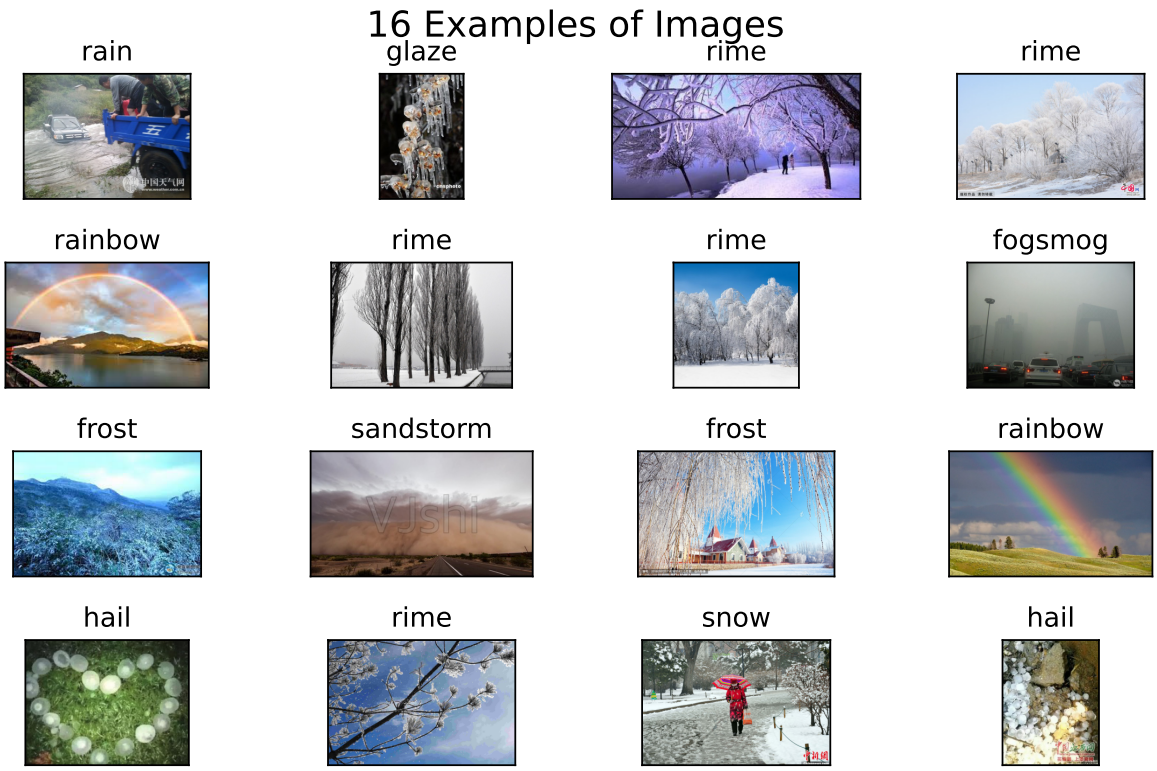
Moving into a continuation of this project the solution would likely be to handle the class imbalance of many more rime images than any other snow/cold related class. This class imbalance could be tackled by creating a weighted loss function to give the classes proportional impact over the weight learning, so underrepresented classes impact the loss more. Another solution could be to augment more images of the underrepresented classes that have issues such as glaze and frost.

More work is to be done to make this a model worth deploying.

Through this assignment I refreshed on many deep learning concepts such as data augmentation and transfer learning. I got to understand the different phases of transfer learning and that batch normalization layers in pre-trained models shouldn't be set to trainable. The main thing I gained was building modular code and class libraries to reuse for the future to make more computer vision tasks easier.

7 Appendix

7.1 Image Examples



7.2 Grayscale Random Forest Results

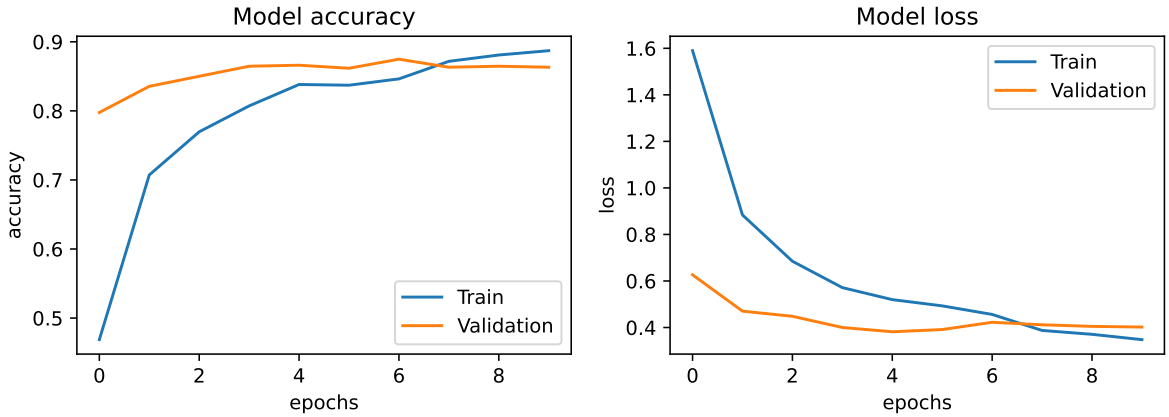
`Classification Report:`

Table 3

	precision	recall	f1-score	support
dew	0.43	0.53	0.47	140
fogsmog	0.55	0.71	0.62	170
frost	0.53	0.21	0.30	95
glaze	0.55	0.44	0.49	128
hail	0.53	0.48	0.51	118
lightning	0.69	0.66	0.68	76

	precision	recall	f1-score	support
rain	0.50	0.35	0.41	105
rainbow	0.75	0.13	0.22	46
rime	0.47	0.66	0.55	232
sandstorm	0.42	0.42	0.42	139
snow	0.58	0.54	0.56	124
accuracy	0.51	0.51	0.51	0
macro avg	0.55	0.47	0.48	1373
weighted avg	0.52	0.51	0.50	1373

7.3 Feature Extraction Results

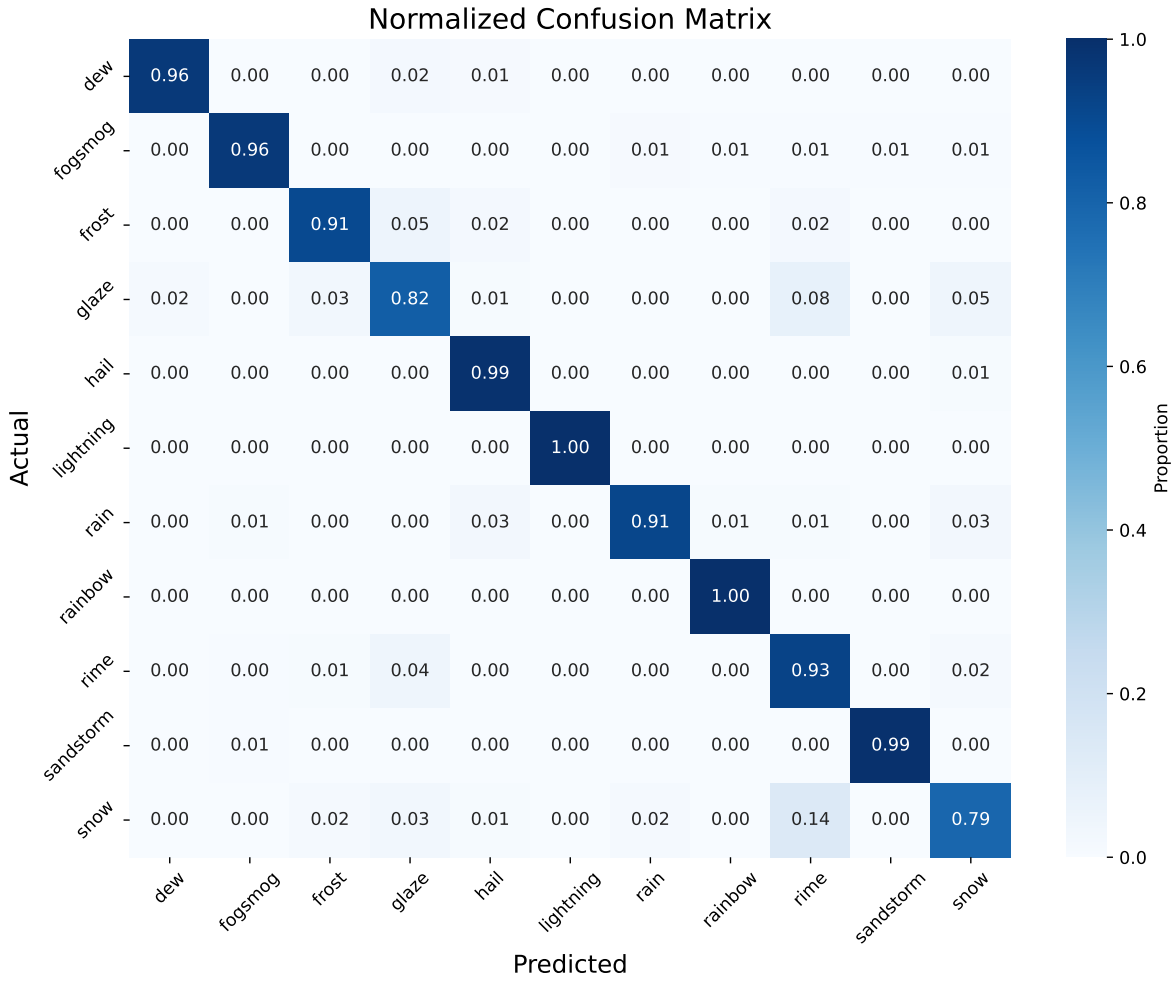


Classification Report:

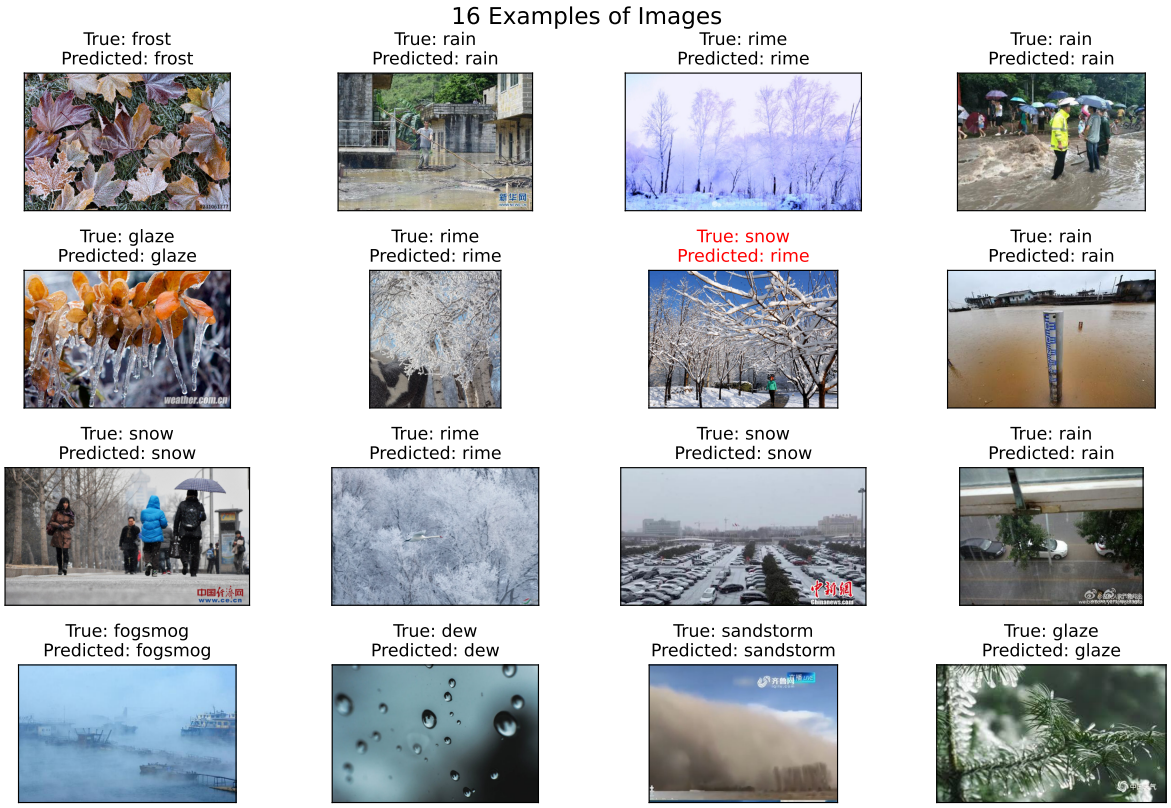
Table 4

	precision	recall	f1-score	support
dew	0.99	0.96	0.97	140
fogsmog	0.98	0.96	0.97	170
frost	0.91	0.91	0.91	95
glaze	0.83	0.82	0.82	128
hail	0.93	0.99	0.96	118
lightning	1.00	1.00	1.00	76
rain	0.96	0.91	0.94	105
rainbow	0.96	1.00	0.98	46
rime	0.87	0.93	0.90	232

	precision	recall	f1-score	support
sandstorm	0.99	0.99	0.99	139
snow	0.87	0.79	0.83	124
accuracy	0.93	0.93	0.93	0
macro avg	0.94	0.93	0.93	1373
weighted avg	0.93	0.93	0.93	1373



<Figure size 3000x2400 with 0 Axes>

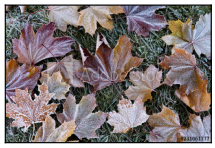


7.4 Fine Tuning Example Images

```
<Figure size 3000x2400 with 0 Axes>
```

16 Examples of Images

True: frost
Predicted: frost



True: rain
Predicted: rain

True: rime
Predicted: rime

True: rain
Predicted: rain

True: glaze
Predicted: glaze

True: rime
Predicted: rime

True: snow
Predicted: snow

True: rain
Predicted: rain

True: snow
Predicted: snow

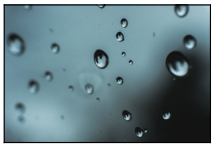
True: rime
Predicted: rime

True: snow
Predicted: snow

True: rain
Predicted: rain

True: fogsmog
Predicted: fogsmog

True: dew
Predicted: dew



True: sandstorm
Predicted: sandstorm

True: glaze
Predicted: glaze