STATS 402 - Interdisciplinary Data Analysis <Weather Classification Through Transfer Learning Based on VGG Models>

<Xiwen Shu, Xuening Peng, Zian Pan>
<xs81@duke.edu, xp27@duke.edu, zp45@duke.edu>

Abstract
Recognizing real-time weather conditions through street-view images has important applications in traffic, traveling, intelligent transportation systems and safety-related research. With the advancement of deep learning theories, convolutional neural networks (CNN) have been universally applied to image recognition and analysis, including automatic weather classification based on images. In this project, we specifically implement a representative model called VGG. VGG improves its configurations by increasing the depth of each hidden layer, which addresses an important aspect of CNNs [1]. The convolutional layers of VGG are used to extract main characteristics of different weather conditions indicated in input images, and then the features extracted at the previous layer become the input of the next layer. The innovative parts of our method include HSV value extraction through image segmentation in the network as additional features and conducting image fragmentation on input images expecting to better predict weather conditions. The dataset we use is a combined dataset gathering images from Weather GAN [2] and Weather Image Recognition [3], containing 341 sunny, 332 rainy, 350 foggy, and 380 snowy street-view labeled images (1403 in total). In terms of model validation, we used validation and test sets to compare and analyze the loss and accuracy after training the models. Therefore, this paper aims to improve the performance of our weather classification model based on VGG by involving HSV values and image segmentation method for real-life applications.

1. Introduction
Dynamic weather conditions have a great impact on society ranging from people's daily life and high-tech applications related to visual conditions influenced by weather. The real-time automatic weather recognition task plays an important role in traffic systems, safety-related topics, intelligent transportation systems and in better understanding cities for policymakers [4]. Firstly, in the modern traffic system, changeable weather conditions would exert a great impact on city transportation systems. Weather like rainy, snowy, and foggy would largely decrease the traffic visibility and friction on the road, which is likely to cause traffic conjestions, life-risking traffic accidents or other potential problems in transportation [5]. Introducing weather image recognition technology on the road could effectively assist in adjusting timing parameters of traffic lights. For example, the system can accordingly extend the time of green/red/yellow light time under severe weather conditions [5], which reasonably maintains the traffic order and lowers the risk to a large extent. Secondly, real-time weather detection can assist some automobile auxiliary systems or autonomous driving systems. For instance, the intelligent driving system could set speed limits in severe weather conditions, provide drivers with immediate reminders of keeping safe distance, and automatically turn on wipers on rainy days [6]. Lastly, monitoring and collecting weather data at streets is also helpful for policymakers in urban traffic/safety regulation-related departments to know the city better and adopt proper strategies to solve potential problems in time.

Traditional weather detection approaches require advanced and expensive sensors installed at multiple places to indicate weathers by collecting and analyzing data of wind speed, humidity, temperature and other weather-related indicators. In addition, the unpredictability and instability of the external environment is likely to affect the recognition accuracy of different sensors, and it is tough to track the dynamic weather conditions by time and space [5]. Instead, by setting cameras on streets or directly using the road monitoring system, we can easily collect the weather information without spatial and temporal limitations.

Our research method in recognizing weather conditions is a transfer learning process based on established VGG deep learning models, which considers HSV value extraction and image fragmentation. As image color can have some implicit indications on different weather, we consider extracting the HSV values of each segmented image and adding these features to the neural network, hoping to achieve higher prediction accuracy. In distinguishing sunny, rainy, snowy and foggy, sunny-day images have a large proportion of blue sky; rainy-day images are clear and with gray color; foggy-day images blurred and with darker gray while snowy-day images tend to have a large proportion of white color. The other improvement is image fragmentation. We divided each street-view image by 2:1:2 to try to fragment the skyline and horizon line of an image so that we can get the color from more targeted regions. This simplifies the process of extracting HSV color values from the whole piece of image with very diverse color properties.

In the subsequent part of this paper, we will introduce the related work done before in detail, propose the VGG+ models we utilized, evaluate the performance of our model through validation and testing, and finally conclude our project and suggest future research direction of it.

2. Related work/literature review
Traditional methods have shown flaws on the installation and maintenance in actual use, not limited to multiple sensors. Alongside the development of intelligent transportation systems,

image-based processing methods including decision tree and support vector machines came into handy in a small range [7]. These methods adopted Regions of Interest extraction, histogram descriptor representative and aforementioned classifiers to accomplish the classification tasks. However, the small size of the dataset results in lack of the robustness of generalizing the model. Meanwhile, only two types of weather can be recognized with the general accuracy being up to 25%, which disables the methods to meet real-life requirements [8].

Recently, deep learning methods have been prevalent and proved to improve previous weather recognition methods. Deep learning allows multi-processing layers of computational models to learn and represent data by emulating how the brain perceives and understands external information, which is good at grasping the connotative connection between large-scale data [9]. Convolutional Neural Networks (CNNs) as one significant method, work well in computer vision, object detection and segmentation within related automatic detection fields. It can capture the nonlinear correlation, discriminate and extract features of each category accurately. In the area of automatic weather classification or detection, existing methods using representative CNNs such as AlexNet, a pretrained model differentiating sunny and cloudy weather dataset, consisted of five convolutional layers followed by three fully connected layers [10]. ResNet is aimed at combining input and output data with the same dimension, which figures out the difference between them, further improving the accuracy [5]. Another salient one, the GoogLeNet, dealing with a large-scale extreme weather dataset, undergoing fine tuning, obtained a more accurate weather identification model with a recognition accuracy of 94.5% based on their constructed "WeatherDataSet" [11]. It is obvious that existing CNNs have great achievements in weather detection, the general accuracy was far higher than traditional methods. But they all require a huge dataset as the backup. The classified categories are also limited, as mentioned before, AlexNet dealing with sunny and cloudy and GoogLeNet with extreme weather. Meanwhile, a large size of parameters is inside the model, thus, high-end CPU and GPUs contribute to expensive training costs and slow pace of training.

Therefore, our model demonstrates a combination of VGG with HSV values for the purpose of pursuing high accuracy and differentiating sunny as well as three easily confused weather: rainy, foggy, and snowy. We specifically choose the VGG model due to its great performance in accuracy. It increases the depth of layers conducive at non-linear mapping and differentiation. The training speed has been largely fastened compared to large ResNet or traditional AlexNet because it uses smaller 3*3 receptive fields with a stride 1 reducing the number of parameters in each layer [12]. Regarding the dataset, we focus more on the street view images rather than random selected weather images for future applications. Instead of only taking CNN into consideration, we combine image segmentation first to obtain color information of each image. Then we choose VGG as one chain of our model and add HSV as important evaluated and supplementary features to finish the model construction.

3. The proposed method

Dataset

Our dataset is a combined dataset gathering 1403 images from Weather GAN [2] and Weather Image Recognition [3], containing 341 sunny, 332 rainy, 350 foggy, and 380 snowy street-view labeled images. Here is the visualization (**Figure 1&2**).

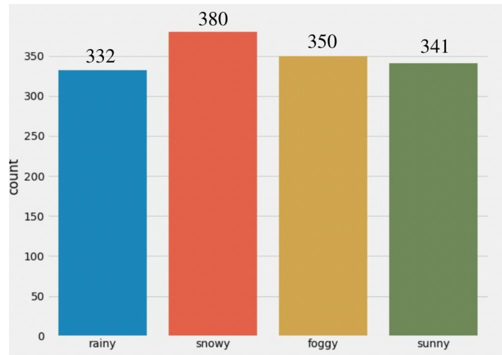

Figure 1. Histogram of dataset

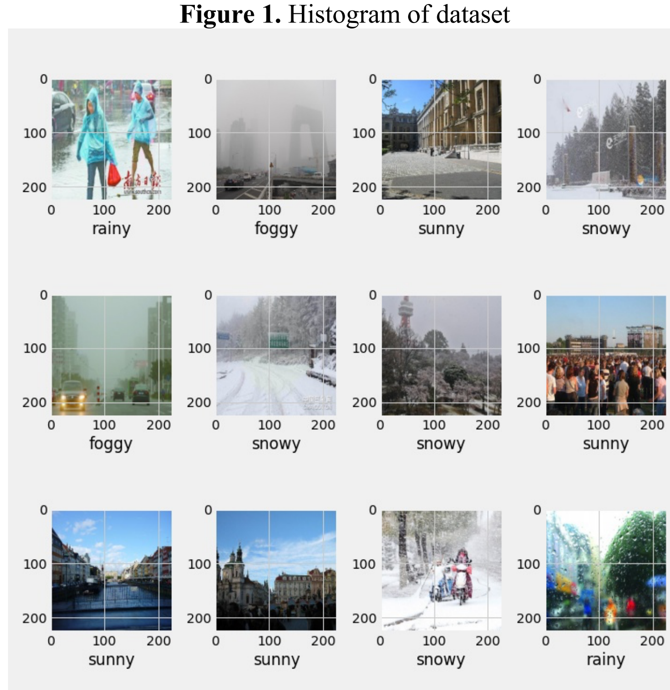

Figure 2. Samples of dataset

Data preprocessing

HSV extraction through image segmentation

As we are dealing with imaging data of weather in outdoor views, one effective approach is to read HSV values from an image. HSV are expanded to hue, saturation, and value, which are alternative representations of the RGB color model. Hue reflects a gradation or variety of a color, by which the color of an object is classified as red, blue, green, or yellow in reference to the spectrum. Saturation is the intensity or purity of color while value indicates the brightness of a color [13]. The combination of the three parameters contribute to the general color reflection of human vision. In order to extract HSV values in a more representative way, we considered implementing image segmentation by K-means algorithm. In image data, this algorithm can help finding subgroups in a given image and assign the pixels to the subgroup. In our method, each image is clustered into 7 color lumps by extracting close BGR color values with k = 7. This procedure helps extract representative color features from an image.

As H (hue) represents different colors, we divided the range into 6 categories, from 0 to 180 with 30 as an interval. Every time we implemented 7-seg image segmentation, we read the extracted H values in each segment and assigned the count of H values that fall into corresponding preset categories. In this case, six H features have been collected. In terms of S and V, as they represent the whole state of each image, we only look at the mean of S and V of each image, which are counted as another two features. Therefore, whenever the HSV extraction through image segmentation is done, we can collect 8 features in total.

Image fragmentation
In addition to considering the color feature of the whole picture, we want to further use image fragmentation to effectively extract colors in the regions of interest. As the input images are street-view pictures, the regions we are interested in are the fragments of sky and ground. So we divided each image into three fragments by 2:1:2, trying to divide it by skyline and the horizon line so that we can get intensive ranges of HSV values centralizing the sky (e.g., blue in sunny) and the ground (white in snowy). After the fragmentation, we conducted HSV through image segmentation again by setting the color clusters for each fragment to k=3. Therefore, we then get 24 features for the fragmented parts in total.

Model architecture
Transferred VGG 19 model
We study the structure and performance of the pretrained VGG 19 model. VGG (stands for visual geometry group), is a pretrained model consisting of 19 layers (16 convolution layers, 5 MaxPool layers, 3 fully connected layers and 1 softmax layer) and implemented in many fields such as transfer learning [14]. It has a 3*3 kernel with stride 1 in each convolutional layer and a 2*2 MaxPooling layer after 2 or 3 convolutional hidden layers, which is regarded as a block. The architecture of VGG 19 is very simple and easy to understand shown in the **Figure 3**.

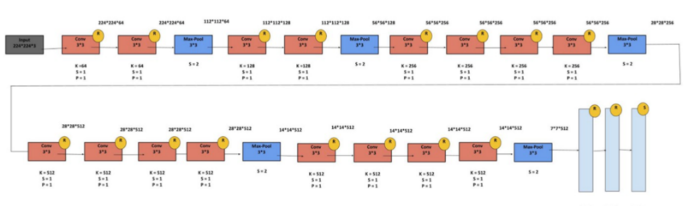

Figure 3. VGG 19 structure

For loss function, weather detection loss is defined by categorical cross entropy. Categorical cross entropy, also called as Softmax loss, is expressed as:

$$\ell(y, \mathbf{p}) = -\sum_{k=1}^{K} q_k \log p_k$$

where $\mathbf{q}$ is the one-hot encoding of y. One-hot encoding is a process by which categorical variables are transformed into a form that can be provided to ML algorithms [15]. For example, $\mathbf{q}$ = [0,0,1,0,0] means category 3 is the true classification. It excels at multi-category classification problems and combines SoftMax output units.

For activation function, ReLu as Rectified Linear unit is expressed as:

$$\rho(a) = \max(0, a)$$

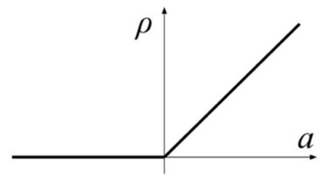

Figure 4. ReLU function

which served as an activation outputting any number bigger or equal to 0.

SoftMax activation is a normalization function, written as:

$$z_k(\mathbf{p}) = \frac{e^{p_k}}{\sum_{j=1}^{K} e^{p_j}}$$

where $\mathbf{p}$ is the output from the network. It rescales the model output to have the correct properties [16]. The sum of all softmax output equals to 1, so the output single vector can be viewed as a probability distribution reflecting the attribution of the input data.

Max-pooling is a downsampling method aimed at retaining the largest output action in each specific square if the layer is regarded as a composition of lots of same sized squares [17]. It chooses the largest value in each square because it is possibly the most salient one representing the whole square and providing more information than other values.

In our model, transfer learning is adopted where knowledge learned from pretrained VGG models is transferred from source dataset to our new dataset. Our model copies all the parameters and designs except for output layers, and we fine tune the parameters and number of layers based on our dataset [14]. The output of pretrained VGG model is a 4*4*512 matrix without the last 4 layers (3 fully connected layers and 1 softmax layer). Thus, we then flatten the matrix to 8192*1 and add three fully connected layers. The first fully connected layer has 512 neurons, second one has 128 neurons and last layer has 64 neurons. All layers have ReLu activation function to finally get a 64*1 output matrix as shown in the **Figure 5**.

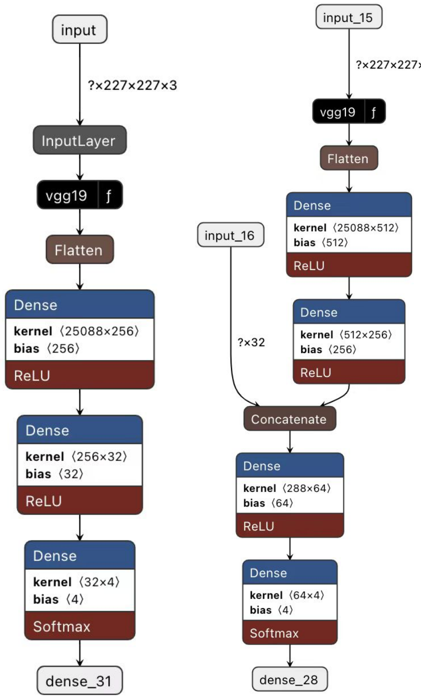

Figure 5. Architecture of Transferred VGG model (left) and HSV-based VGG model (right)

HSV-based VGG model

As mentioned, we have the general image information: 8 features collected from one image, and another 8*3 features from fragmented three parts in each image. Therefore, after flattening, we get a 32*1 matrix representing HSV features in each image. A 32*1 matrix of HSV values is then combined with 64*1 output from VGG model into the 96*1 matrix. We further take a 96*1 matrix as the input of the second designed neural network. The following neural network is constructed by tuning parameters and number of layers or neurons. Finally, it is made up of three fully connected layers. The first layer has 32 neurons, second one has 16 neurons and the last layer has 4 neurons respectively using Relu, Relu activation function and softmax to and produce a 4*1 matrix representing the four weather categories.

For simplicity, we call the model *HSV-based VGG model* In conclusion, this model contains VGG, three fully connected layers (512, 128, and 64 neurons) and the second designed network. And the second network takes the output from VGG of 64 neurons, adding the 32 HSV values as its first layer (96 neurons), following a layer of 32 neurons, 16 neurons and the last layer of 4 neurons respectively (**Figure 5**).

4. Performance evaluation

Experiment

The dataset is first divided into three sets. Around 1.1 thousand pictures are used for training, 2.5 hundred for testing and 2 hundred for validation. These data are then fed to both transferred VGG (pretrained VGG 19 + 3 dense layers, see **Figure 5** right) and our HSV-based VGG model (see **Figure 5** left) for training. In terms of model construction, we take advantage of keras functional api, which is flexible for multiple inputs. In the training process, we feed the BGR array of images to both transferred VGG and HSV-based VGG. We also feed 32*1 hsv features of each image to HSV-based VGG model as a second input. During training, the batch size is set to 10, 20, 50, 80, 100 respectively for each model and hyperparameter patience is set to 10, which means the program will stop if the validation loss does not increase in 10 epochs to avoid overfitting. After that, we will use different metrics such as accuracies, loss, recall and F1-score to evaluate the performance of our models.

Results and Discussion

After running the experiments several times by tuning parameters, HSV-based VGG model reach its best performance at batch size equal to 20, while Transferred VGG model equals to 100. The recognition accuracy of each weather condition is calculated, showing in the confusion matrix (**Figure 6**). The values on the diagonals represent the recognition accuracy of each category, respectively. For the transferred VGG models, the accuracy of rainy days is 89%, that of snowy days is 83%, that of foggy days is 95%, and that of sunny days is 93%. For the HSV-based VGG models, the accuracy of rainy days is 93%, that of snowy days is 86%, that of foggy days is 95%, and that of sunny days is 95%. The average accuracy of transferred VGG is 89.50% while that of HSV-based VGG is 91.60%.

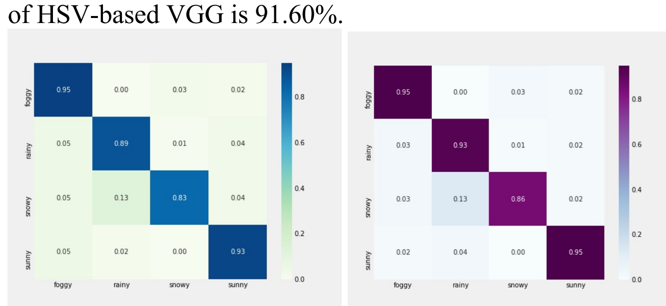

Figure 6. Confusion matrices of transferred VGG (left) and HSV-based VGG (right)

There is also a visualization of overall performance of two models. **Table 1** shows the classification report of each model at the testing set. The transferred VGG model has an accuracy of 89.50% and 0.31 loss while HSV-based VGG reaches an accuracy of 91.60% with loss of 0.32. There is a slight advantage of HSV-based models, with 2.1% higher accuracy and 0.01 lower loss. The HSV-based VGG models also show higher precision, recall and F1-score, which indicates a lower false positive rate in classification.

Experiments	Accuracy (%)	Loss	Precision	Recall	F1-score
Transferred VGG	89.50	0.31	0.895	0.9	0.89
HSV-based VGG	91.60	0.32	0.9175	0.9225	0.92

Table 1. The classification report of each model for the test sets

As we can see from **Figure 7**, the validation accuracy of HSV-based VGG model achieves the highest when epoch = 7 which is

91.60%. Also, we check the loss of the training model when epoch = 7, the validation loss is back to 0.32 where we save the best for our final training model. Because we set patience = 10, the model keeps checking for another 10 epochs without finding a possible higher accuracy until it meets the requirement of "early stopping" and stops.

And for Transferred VGG model in **Figure 8**, it achieves the highest when epoch = 9 which is 89.50%. Also, we check the loss of the training model when epoch = 9, the validation loss is back to 0.31 where we save the best for our training model. Because we set patience = 10, the model keeps checking for another 10 epochs without finding a possible higher accuracy until it meets the requirement of "early stopping" and stops.

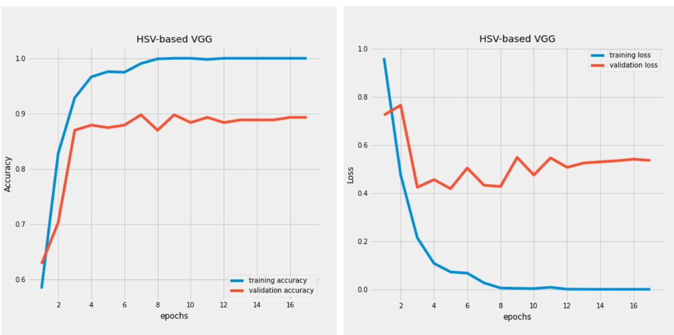

Figure 7. The training and test accuracies/loss for HSV-based VGG model

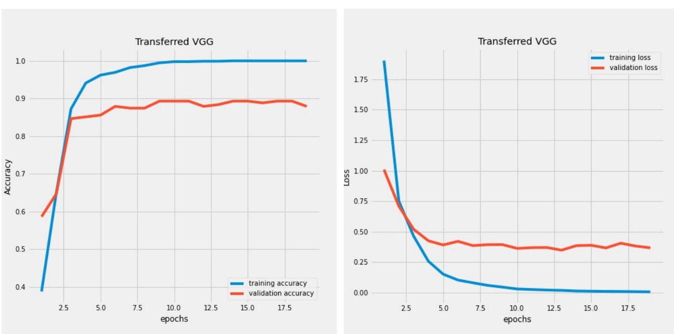

Figure 8. The training and test accuracies/loss for Transferred VGG model

The color feature of an image has been proved to be effective in recognizing different weather conditions. The reason for why HSV-based models could perform better than the original ones could be our innovative approach to extract HSV values of an image. We not only extract these values through K-means clustering in image segmentation, but also consider conducting image fragmentation to divide out sky and ground for more accurate color extraction. These all contribute to the better performance of HSV-based models. However, the advantage is still not remarkable which might also be the result of HSV extraction. The method we invented is to consider the detailed categories of H values only and calculate mean values of S and V values. This method can help highlight the importance of color variety but may ignore some important features hidden in S and V values. A more scientifically proposed calculated method should be further explored and developed through experiments. Overall, involving the color feature in weather image recognition can be a valuable research direction to consider in future related works.

5. Conclusion and future work

In conclusion, in order to accomplish automatic weather image recognition on street-view pictures, we trained on the combined dataset using HSV-based VGG networks, and finally improved the performance of our weather classification model compared to the original one. The strategies we used, including extracting HSV values in a tailored way by image segmentation and image fragmentation play an important role in extracting representative features of each image aside from the pretrained VGG models. Therefore, although there is only a slight improvement in model accuracy from transferred VGG to HSV-based VGG, it is still significant to involve color features of an image to help identify the weather condition. These considerations can be hopefully included in the applications of automatic weather image detection after balancing the gain and loss. In the following work, we can still modify our model and try to reduce the layers because even if the accuracy is high it still takes up a high GPU and CPU. Regarding the dataset, our model classifies four types of weather, therefore, we can include other types of weather and make our model more generalized and robust. Also, the dataset we use contains a total around 1400 photos which is a relatively small dataset and all the images are taken at similar angles, thus more different kinds of images require to be investigated to testify the feasibility of our model. What's more, our images are all taken in the daytime, it is helpful to extend the weather conditions recognition to night time.

References

[1] J. Wei, "VGG Neural Networks: The Next Step After AlexNet," *Medium*, Jul. 04, 2019. https://towardsdatascience.com/vgg-neural-networks-the-next-step-after-alexnet-3f91fa9ffe2c (accessed Mar. 04, 2022).

[2] "Weather GAN?" https://kaggle.com/mpwolke/weather-gan (accessed Mar. 04, 2022).

[3] "Weather Image Recognition." https://kaggle.com/jehanbhathena/weather-dataset (accessed Mar. 04, 2022).

[4] M. R. Ibrahim, J. Haworth, and T. Cheng, "WeatherNet: Recognising weather and visual conditions from street-level images using deep residual learning," *ArXiv191009910 Cs*, Oct. 2019, Accessed: Mar. 06, 2022. [Online]. Available: http://arxiv.org/abs/1910.09910

[5] J. Xia, D. Xuan, L. Tan, and L. Xing, "ResNet15: Weather Recognition on Traffic Road with Deep Convolutional Neural Network," *Adv. Meteorol.*, vol. 2020, p. e6972826, May 2020, doi: 10.1155/2020/6972826.

[6] Y. Shi, Y. Li, J. Liu, X. Liu, and Y. L. Murphey, "Weather Recognition Based on Edge Deterioration and Convolutional Neural Networks," in *2018 24th International Conference on Pattern Recognition (ICPR)*, Aug. 2018, pp. 2438–2443. doi: 10.1109/ICPR.2018.8546085.

[7] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar, and S. Mishra, "Decision Tree and SVM-Based Data Analytics for Theft Detection in Smart Grid," *IEEE Trans. Ind. Inform.*, vol. 12, no. 3, pp. 1005–1016, Jun. 2016, doi: 10.1109/TII.2016.2543145.

[8] S. G. Narasimhan, C. Wang, and S. K. Nayar, "All the Images of an Outdoor Scene," p. 15.

[9] "Deep Learning for Computer Vision: A Brief Review." https://www.hindawi.com/journals/cin/2018/7068349/ (accessed Mar. 10, 2022).

[10] C. Lu, D. Lin, J. Jia, and C.-K. Tang, "Two-Class Weather Classification," 2014, pp. 3718–3725. Accessed: Mar. 10, 2022. [Online]. Available:

https://openaccess.thecvf.com/content_cvpr_2014/html/Lu_Two-Class_Weather_Classification_2014_CVPR_paper.html

[11] "Extreme Weather Recognition Using Convolutional Neural Networks | IEEE Conference Publication | IEEE Xplore." https://ieeexplore.ieee.org/document/7823700 (accessed Mar. 10, 2022).

[12] "A practical experiment for comparing LeNet, AlexNet, VGG and ResNet models with their advantages and disadvantages. | by Tejas Mohan Ayyar | Medium." https://tejasmohanayyar.medium.com/a-practical-experiment-for-comparing-lenet-alexnet-vgg-and-resnet-models-with-their-advantages-d932fb7c7d17 (accessed Mar. 10, 2022).

[13] "HSL and HSV," *Wikipedia*. Mar. 02, 2022. Accessed: Mar. 07, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=HSL_and_HSV&oldid=1074794216

[14] "13.2. Fine-Tuning — Dive into Deep Learning 0.17.4 documentation." https://d2l.ai/chapter_computer-vision/fine-tuning.html (accessed Mar. 10, 2022).

[15] "What is One Hot Encoding? Why and When Do You Have to Use it? | HackerNoon." https://hackernoon.com/what-is-one-hot-encoding-why-and-when-do-you-have-to-use-it-e3c6186d008f (accessed Mar. 10, 2022).

[16] "Multi-Class Neural Networks: Softmax | Machine Learning Crash Course," *Google Developers*. https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax (accessed Mar. 10, 2022).

[17] J. Brownlee, "A Gentle Introduction to Pooling Layers for Convolutional Neural Networks," *Machine Learning Mastery*, Apr. 21, 2019. https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/ (accessed Mar. 10, 2022).