Image Classification of Weather - Dew, Fog/Smog, Frost, Glaze, Hail, Lightning, Rain, Rainbow, Rime, Sandstorm and Snow

Jessica Lopez, Nazmul Shahadat
Data Science, Truman State University
Kirksville, Missouri, USA
ef16538@truman.edu

***Abstract*—This study focuses on the recognition of weather patterns in images. Using a dataset of different weather patterns, a model can learn to detect and properly trace the type of weather that occurs in each image. This type of recognition model can be used to predict upcoming weather patterns or the level of storm severity.**

***Index Terms*—image-recognition, weather, machine-learning, prediction**

I. Introduction

* This first pre-analysis was originally conducted before (yet later edited succeeding) image classification testing to understand and set proper hypophysis questions and solution goals and without bias.

Initial-Hypotheses

1) The machine learning model performs significantly better than random guessing in classifying weather conditions.

2) Humidity and cloud cover do not significantly affect the model's ability to detect rainfall.

About-the-Dataset

The initial data set is titled "Weather Image Recognition." It contains 11 folders of images (jpg). The folder classes are dew, fog/smog, frost, glaze, hail, lightning , rain, rainbow, rime, sandstorm and snow. The largest folder set is rime with 1160 images and the smallest data set is rainbow with 232 images. This data set was pulled from Kaggle.com and it was published by Jehan Bhathena, a student at VIT Vellore, last updated 5 years ago in 2021.

II. Methodology

All data analysis, models, and visualizations will be derived from Python and R code. An understanding of data models and image detection will be needed for this study.

Clarity and Focus - The initial data set is clean and does not need further preparation since it involves only images. There are no missing values or text errors.

Readability - Yes, the results of the model can be replicated and trusted.

Validity - Each research question or hypothesis can be answered using this model. The data is correct and valid.

Transparency - The outcome and final results can be evaluated and shared to enhance credibility and information use.

Adaptability - If there is a change in the future to this study, it can be updated and re-marked to that date and year.

III. Expected-Outcomes

I expect the model to correctly identify the images after processing and be better at identifying image weather patterns than random guessing.

A. Further-Research

- Medium Articles on Weather Image Detection: Source: https://urvog.medium.com/weather-image-classification-with-keras-4eee9468ff2f
- IEEE-Xplore Weather Classification for Multi-class Weather Image Based on CNN Source: https://ieeexplore.ieee.org/document/9943190

IV. Conclusion

I found from this exercise that using data science methodology and python programming, that you can construct a strong weather study practice by working with image detection based models. I would now expect in using similar patterns, that interpreting other data sets would provide similar success in further studies in other areas. For example, animal or insect detection or brain/bone scanning images.

Initial-Images

Examples of a few images from the original data set.

Fig. 1: Image from the Dew data set.

Fig. 2: Image from the Fogsmog dataset.

Fig. 3: Image from the Rain dataset.

Fig. 4: Image from the Lightning dataset.

Fig. 5: Image from the Rainbow dataset.

Fig. 6: Image from the Snow dataset.

Fig. 7: Image from the Frost dataset.

Fig. 8: Image from the Glaze dataset.

Fig. 9: Image from the Hail dataset.

Fig. 10: Image from the Sandstorm dataset.

Fig. 11: Image from Snow dataset - 1 Mountains View.

Fig. 12: Image from Snow dataset - 2 Forest Trees.

Cite:(1,Kaggle)

Python-Image-Classification-Model

* This is the end of pre-analysis. The results and solutions from the image classification testing start here.

Implementation

Classes:	Dew, Fogsmog, Frost, Glaze, Hail, Lightning, Rain, Rainbow, Rime, Sandstorm, Snow
Number of classes:	11
Total images:	6862

TABLE I: Initial Data Set Count

Training images:	5489
Validation images:	1373

TABLE II: Data Set Count after Image Classification Training

The model prepares a weather image dataset for image classification using PyTorch. The dataset is stored in a main folder called dataset, with each weather condition stored in its own subfolder. The ImageFolder function automatically uses the folder names as class labels, allowing the model to classify images into categories such as dew, fogsmog, frost, glaze, and rainbow.

Before training, each image is resized to 150 by 150 pixels, converted into a tensor, and normalized so the pixel values are scaled consistently. The dataset is then divided into training and validation sets using an 80/20 split. The training set is used to teach the model, while the validation set is used to evaluate how well the model performs on images it has not trained on.

Overall, this code does not yet train the image classification model, but it correctly prepares the dataset for training. The next step would be to define a neural network model, choose a loss function and optimizer, train the model over several epochs, and evaluate its accuracy on the validation data.

Image-Comparison

The images within each dataset have many similar and contradicting factors. The model needs to be able to define the correct class for each image even with the differences in styles with each dataset.

Fig. 8 and Fig. 9 are from the snow data set. The images need to be detected as snow weather despite the confusion of the different colors, time of day and settings.

Further-Breakdown-of-Information-of-Training-Data

The libraries used with this model are PyTorch, Scikit and matplotlib.pyplot.

The directory of the data set is "./dataset".

Every image is resized and standardized into a 150 by 150 pixel jpg image.

The machine learning model splits the data into training and validation sets.

This uses 80 percent of the images for training and 20 percent for validation.

After being run through an editor, the Python code shows the results in Tables I and II.

2026-Current-Definition-of-Image-Classification

Image Classification is a machine learning term for grouping elements or images into predefined groups or categories based on recognition.

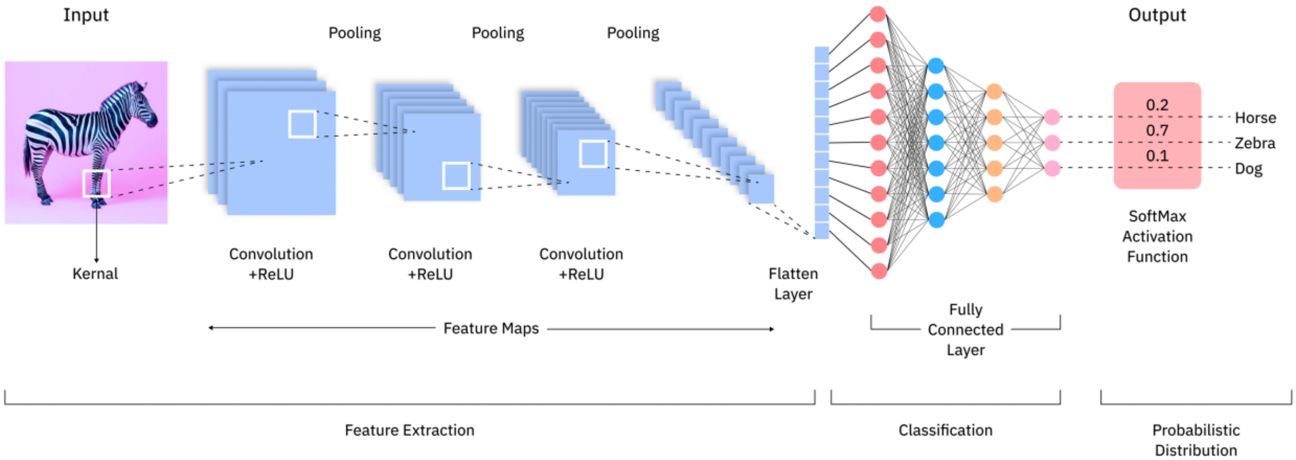

Fig. 13: IBM-Diagram of a Convolutional Neural Network (CNN)

Cite: (2, IBM-What-is-Image-Classification?)

This term was originally approached with the aspect of children that play with blocks, plants or animals and how they sort them into different piles based on color and features.

AI (Artificial Intelligence) has mimicked the children by sorting through image and text based models. With statistical image classification, it takes on this task by training the models to recognize patterns that are embedded in the images.

As of July 2026, the use of Convolutional Neural Networks (CNNs) are the most recent major breakthrough in the field of Artificial Intelligence (AI).

The evolution of image classification has moved from handcrafted feature engineering to deep learning, then to pretrained models and transformer-based architectures, and now toward large multimodal foundation models. These advances have significantly improved accuracy, reduced the need for large labeled datasets through transfer and self-supervised learning, and expanded image classification into systems that can reason about visual content using natural language. As a result, image classification is now more accurate, adaptable, and applicable.

Some industrys that have recently increased usage of Image classification is healthcare, autonomous driving, agriculture and environmental monitoring.

Formula for Conditional Probability Distribution

$$(X|Y) = \frac{P(XandY)}{(P(X)}$$

P = Probability distribution of X given Y

X = Probability that one variable is true.

Y = Probability that a second variable is true.

Machine-Learning-and-Deep-Learning

Machine learning and deep learning are both methods used for image classification, but they differ in how they learn and process image data. Traditional machine learning models require humans to identify and select important features from images, such as edges, colors, textures, or shapes, before the model can classify them. For example, a machine learning system designed to recognize animals may need manually created features that describe differences between cats and dogs. The model then uses these features to classify new images.

Machine learning is a branch of artificial intelligence that allows computers to learn from data and improve their performance without being explicitly programmed for every task. Instead of following a fixed set of instructions, machine learning algorithms identify patterns and relationships within data to make predictions, classify information, or make decisions. The more relevant data a model receives, the better it can become at recognizing patterns and producing accurate results.

Deep learning is a branch of artificial intelligence and machine learning that uses artificial neural networks with multiple layers to learn from large amounts of data. These layers allow deep learning models to automatically identify patterns, extract important features, and make predictions without requiring humans to manually program specific rules. The word “deep” refers to the many layers within the neural network that process information at increasing levels of complexity.

Further-Explanation-of-Neural-Networks

A neural network is a type of artificial intelligence model inspired by the way the human brain processes information. It consists of interconnected layers of artificial neurons that receive, process, and transmit data. During training, a neural network learns by analyzing large datasets and adjusting the strength of connections between neurons to improve its predictions. This learning process enables the model to recognize complex patterns and relationships that may not be obvious through traditional programming.

Neural networks were used in this image classification, where the goal was to identify and categorize objects within an image. By training on labeled images, the network learns to detect important visual features such as edges, shapes, textures, and colors. As data moves through multiple layers, the network combines these features to recognize more complex objects, such as weather patterns.

This technology powers applications that make image classification faster, more accurate and highly efficient.

Neural-Network-Compassion-to-Brain

-The human brain is made of billions of neurons while neural-networks are made of artificial "neurons" (nodes).

-Neurons connect through synapses and nodes connect through weighted links.

-A brain can receive information through senses while a network receives input data.

-Learns by strengthening or weakening synapses while networks learn by strengthening or weakening synapses.

-It produces thoughts, decisions and actions similar to a network that produces predictions, classifications, or generated content.

Different-Types-of-Image-Classification

Binary-Classification This involves classifying images into one of two categories. For example, determining whether an image contains a cat or not. It's often used in medical imaging (disease vs. no disease) or defect detection in manufacturing.

Multiclass-Classification Here, images are categorized into more than two classes. For instance, classifying animal images into "cat," "dog," or "horse." Each image belongs to exactly one class.

Multilabel Classification An image can be associated with multiple labels simultaneously. For example, a photo could be tagged as both "beach" and "sunset." This is common in scene recognition and tagging systems.

Hierarchical-Classification Classification occurs at multiple levels of a hierarchy. For example, an image might first be classified as "mammal" and then further as "cat" or "dog." This is useful in taxonomy-based datasets.

Rule-Based-Classification Relies on manually defined rules and features (e.g., template matching, thresholding) to classify images. It's interpretable and customizable but less adaptable to complex patterns.

Statistical-Classification Uses algorithms like SVM, Random Forest, or KNN to learn from labeled datasets. These methods can be distribution-based (e.g., Maximum Likelihood Estimation, Bayesian classifiers) or distribution-free (e.g., CNNs) for feature learning.

Deep-Learning-Based-Classification Employs Convolutional Neural Networks (CNNs) such as AlexNet, GoogLeNet, or ResNet to automatically learn hierarchical features from raw pixels. Often combined with transfer learning for efficiency and accuracy.

–

For this study, I used Multiclass-Image-Classification through a weather image based data set. The model detected the images and recognized each images grouping.

Understanding-Image-Classification

Future studies of similar topics or compassion among other data sets or other data models is also good for better accuracy and credibility among machine learning models.

One topic that is similar is "cars" that look for tires, doors and side-view mirrors.

Another example is "dog" identification that checks for floppy ears, wagging tails and long noses.

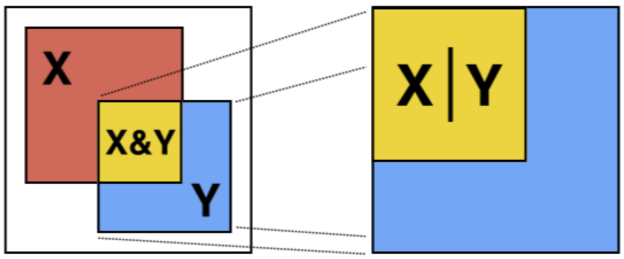

Fig. 14: Probability Distribution Diagram, Cite: (4, Brilliant-Conditional Probability Distribution)

Also, “Lake” classification can find photos with lots of water and shoreline.

I also find that leaf or plant identification could be used with Image classification.

About-Python-Coding-Language

For this assignment I used only Python3 in July, 2026.

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. Created by Guido van Rossum and first released in 1991, Python has become **”the world’s most popular programming language”**. Its clean syntax allows beginners to learn programming concepts quickly while also providing advanced features that experienced developers can use to build complex applications. Python supports multiple programming paradigms, including object-oriented, procedural, and functional programming.

One of Python’s greatest strengths is its extensive standard library and large ecosystem of third-party packages. Developers use Python for web development, data analysis, artificial intelligence, machine learning, scientific computing, automation, cybersecurity, and game development. Popular frameworks such as Django and Flask simplify web application development, while libraries like NumPy, Pandas, and TensorFlow support data science and machine learning projects.

Python is also highly portable. Its active global community continuously develops new tools, tutorials, and resources, making it easy for learners to find support. Because of its flexibility, ease of use, and wide range of applications, Python is widely taught in schools and universities and is frequently chosen by businesses and researchers for solving real-world problems efficiently.

Libraries-and-Frameworks-Used

I used the following Python frameworks for this image classification to have the fastest and cleanest code possible.

Pathlib manages the data, PyTorch learns from the images, and Matplotlib.pyplot helps visualize and interpret the results.

Pathlib - Object-oriented filesystem paths

Pathlib is a built-in Python module that provides an object-oriented way to work with file and directory paths. It was made to simplify path handling and replace many functions from the os and os.path modules. When using the Path class, I can create, combine, rename, and manage file paths with clear and readable syntax. Pathlib also supports checking file existence, creating directories, reading and writing files, and searching for files with pattern matching. It automatically handles differences between operating systems, making code more portable, reliable, and easier to maintain in modern Python applications.

PyTorch

PyTorch is an open-source Python-based machine learning framework developed by Meta for building and training deep learning models. It provides flexible tools for creating neural networks, performing tensor computations, and using automatic differentiation to optimize model training. PyTorch is widely used in research and industry because of its dynamic computation graph, which makes debugging and experimentation easier. It supports CPU and GPU acceleration, enabling efficient processing of large datasets and complex models. The framework also includes libraries for computer vision, natural language processing, and reinforcement learning, making it a powerful choice for developing modern artificial intelligence and deep learning applications.

Matplotlib.Pyplot

matplotlib.pyplot is a module in the Matplotlib library used for creating a wide variety of charts, graphs, and visualizations in Python. It provides a simple interface for plotting data, scientific computing, and machine learning. With matplotlib.pyplot, users can create line graphs, bar charts, histograms, scatter plots, pie charts, and more. It also allows customization of titles, labels, colors, legends, and axes to improve presentation. The module works well with libraries such as NumPy and Pandas, enabling users to visualize data effectively and communicate insights through clear, informative graphical representations.

About-the-OS-(Operating-System)

This study and implementation was run on a MacBook Pro. It is a 14-inch silver model. It has a chip of Apple M3 Pro and a memory of 18 GB. The software version

```
Classes: ['dew', 'fogsmog', 'frost', 'glaze', 'hail', 'lightning', 'rain', 'rainbow', 'rime', 'sandstorm', 'snow']
Epoch [1/10], Loss: 1.4182, Train Accuracy: 53.47%, Validation Accuracy: 65.62%
Epoch [2/10], Loss: 1.0833, Train Accuracy: 63.45%, Validation Accuracy: 66.28%
Epoch [3/10], Loss: 0.9428, Train Accuracy: 68.21%, Validation Accuracy: 67.95%
Epoch [4/10], Loss: 0.8280, Train Accuracy: 72.24%, Validation Accuracy: 72.61%
Epoch [5/10], Loss: 0.6483, Train Accuracy: 77.63%, Validation Accuracy: 72.98%
Epoch [6/10], Loss: 0.5302, Train Accuracy: 81.85%, Validation Accuracy: 71.52%
Epoch [7/10], Loss: 0.4090, Train Accuracy: 86.15%, Validation Accuracy: 72.10%
Epoch [8/10], Loss: 0.3379, Train Accuracy: 88.43%, Validation Accuracy: 73.27%
Epoch [9/10], Loss: 0.2979, Train Accuracy: 90.31%, Validation Accuracy: 73.34%
Epoch [10/10], Loss: 0.2260, Train Accuracy: 92.49%, Validation Accuracy: 73.12%
```

Fig. 15: Model's Probability Distribution Results (From Console)

is up to date as of July 2026 and is at 26.5.2. No system errors or Malware were found during implementation.

Breakdown-of-Model-Python-Code

At first, PyTorch and other Python libraries were imported. The data set was then loaded and assigned as a data directory.

The model was then created and defined by the training set. The parameters were set to the loss function and optimized. The model was then split into trained and validation sets.

The model was then created with a loss function and optimization and then trained. Validation accuracy was then tested.

The final results was a print of Epoch, Loss function, Training accuracy and Validation accuracy of 1/10-10/10 (Fig 9).

Model-Results

The results showed that the neural network is learning the training data well, but it also showed signs of overfitting after several epochs.

As a positive, the loss decreased through each run. A lower loss means the model's predictions are became closer to the correct weather classes.

Validation accuracy also improves, then levels off.

The difference between training accuracy and validation accuracy is 19.37 percentage points. The large gap shows that the network fits the training images much better than unseen images (overfitting).

The best performing model was 9 at 73.34 percent validation accuracy.

Hypotheses - The validation accuracy of random guessing with this model is about 1/11 or 9 percent. The model after running and tuning resulted in a score of about 73 percent, which is far above random guessing. So yes, the first hypothesis was proven accurate because the model performed much better than random guessing. With the second

hypothesis, it was not affected by factors and features like cloud cover and rainfall. It did not seem to affect the models ability to recognize images proving the hypothesis accurate.

Qualitative-and-Quantitative-Data

This Image classification model took qualitative data (jpg images) and resulted in quantitative data of epoch, loss, train accuracy and validation accuracies.

The classes output is qualitative data that can not be sorted by order, while the rest of the output data can be analysed by order/scale as testing results.

Nominal-and-Ordinal-Data

Classes are nominal data sense it is sorted by catagroy. The rest of the outputed data is Ordinal since it is numerical and can be ordered by percentage.

Softwares-and-Tools-Used

Visual-Studio-Code

Visual Studio Code (VS Code) is a free, lightweight source code editor developed by Microsoft. VS Code offers features such as syntax highlighting, code completion, debugging, version control integration with Git, and a built-in terminal. Its large marketplace provides extensions that add support for additional languages, themes, and tools, making it highly customizable. Its speed, user-friendly interface, and cross-platform compatibility make it one of the most popular code editors available today.

Jupiter-Notebook-Visual-Studio-Installation

Fig. 16: Images from a dog data set (Cite 3, Keras)

Jupyter Notebook is an open-source web-based application that allows users to create and share interactive documents containing live code, text, equations, visualizations, and charts. It is widely used for data analysis, machine learning, scientific research, and education because it enables users to execute code one section at a time and immediately view the results. Jupyter Notebook supports using code chunks along side markdown for better documentation. Its interactive environment makes it easier to test ideas, document workflows, and present findings. This flexibility has made Jupyter Notebook a popular tool among developers.

Overleaf-Online-LaTeX-Editor

Overleaf is a cloud-based online LaTeX editor that allows users to create, edit, and collaborate on professional documents directly in a web browser. It is widely used for writing research papers, theses, technical reports, resumes, and academic assignments because it produces high-quality typeset documents. Overleaf provides real-time collaboration, automatic saving, version history, and access to thousands of templates for different document types. Users do not need to install LaTeX software locally, making it convenient for beginners and experienced authors alike. Its integration with reference management tools and Git also simplifies document preparation and collaboration for academic and professional projects.

Note-about-Predictions

This study is meant to recognize images. Image classification, in all cases, does not predict the future. The weather images from this data set do not have any time associations. The time when each picture was took is unknown. Using the data set to make predictions about future weather is not part of this study and is generally not something that can be done with image classification alone. This testing is more for recognition of patterns among AI (artificial intelligence) and machine learning.

Similar-Real-World-Models

This use of image classification focused on the detection of weather images. Similar models with different data sets can be developed for identification purposes. Usually healthcare care and prediction data use image classification to identify types of disease or trends in a companies analysis report. More examples of real-world image classification data sets are:
-Animal classification, example dog (3, Keras)
-Fish or aquatic life classification (6, Viam)
-Plant identification (5, Lightly AI)
-Body, brain or eye scans for detection of diseases and other related health issues
-Autonomous navigation (6, Viam)
-Quality assurance (6, Viam)
-Face and person recognition (6, Viam)
-Traffic flow management (6, Viam)

Conclusion

Overall, these training results indicated this was a successful image classification model with strong learning behavior, while also showing opportunities to improve its performance by reducing overfitting and increasing its ability to generalize weather images. This training model provided over 70 percent accuracy with trial 10 at over 90 percent. The loss function recessively decreased with each trial showing that it became closer to prediction and true values. Validation Accuracy increased sightly continuously leading to better accuracy recognition rating results.

Sources-and-Resources

1) Kaggle.com-Weather-Data-Set, www.kaggle.com/datasets/jehanbhathena/weather-dataset?
2) IBM-What-is-Image-Classification?, www.ibm.com/think/topics/image-classification.
3) Keras-Image-Classification-From-Scratch, keras.io/examples/vision/image˙classification˙from˙scratch.
4) Brilliant-Conditional-Probability-Distribution, brilliant.org/wiki/conditional-probability-distribution.
5) Lightly AI - Image Classification: Types, How It Works, Applications Challenges, www.lightly.ai/blog/image-classification
6) Viam - Image classification in machine learning: A full guide (2024), www.viam.com/post/guide-to-image-classification-in-machine-learning-2024