

Term Project

Deep Learning
2025 spring

Datasets

- Describable Textures Dataset

- The Describable Textures Dataset (DTD) is an image classification dataset based on a diverse collection of texture images captured in the wild. It focuses on describing visual textures using meaningful attributes and is used to train models for recognizing and classifying texture patterns. - Describing Textures in the Wild¹ -
- Input size: various sizes (color images) → Images need to be resized for training.
- Classes: 47 total → Select only 20 classes for term project.
 - Banded/ bubbly/ braided/ cracked/ chequered/ dotted/ flecked/ frilly/ grid/ knitted/
 - lacelike/ marbled/ meshed/ paisley/ pleated/ porous/ scaly/ spiralled/ swirly /zigzagged

- Data size: 2,400

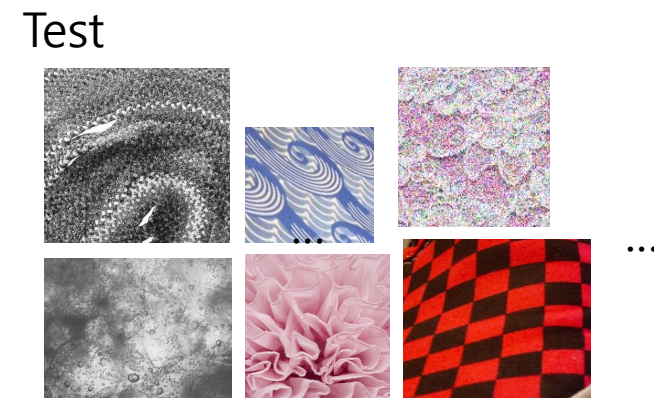


Figure 2: The 47 texture words in the **describable texture dataset** introduced in this paper. Two examples of each attribute are shown to illustrate the significant amount of variability in the data.

¹Cimpoi, Mircea, et al. "Describing textures in the wild." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

Goal

- Apply the concepts learned in class to train a neural network with good **generalization performance!**
- Test data
 - Contains the same 20 classes as in the provided data.
 - **The provided data and test data may not share the same data distribution (cropping, grayscale, noise..)**
 - Not disclosed to students



Test Process

- Skeleton code will be provided
- Execute run.py to generate a result.txt file
 - This file will contain the model's predictions on the test set.
- Your test accuracy will be evaluated based on this output. (accuracy)

```
if __name__ == '__main__':  
    parser = argparse.ArgumentParser(description='2023 DL Term Project')  
    parser.add_argument('--load-model', default='checkpoints/model.pth', help="Model's state_dict")  
    parser.add_argument('--batch-size', default=16, help='test loader batch size')  
    parser.add_argument('--dataset', default='test_images/', help='image dataset directory')
```

- Make sure to **specify the path to your trained model** in the arguments of run.py.

Skeleton code

- Term_Project
 - checkpoints
 - data
 - 0
 - 1
 - ...
 - utils
 - _utils.py
 - model.py
 - test.py
 - train.py
 - run.py

execute run.py



- Term_Project
 - checkpoints
 - data
 - 0
 - 1
 - ...
 - utils
 - _utils.py
 - model.py
 - test.py
 - train.py
 - run.py
 - result.txt



result.txt ✕	
1	0
2	0
3	0
4	0
5	0
6	2
7	0
8	0
9	0
10	0
11	1
12	1
13	1
14	1
15	1
16	1
17	1

Before submission, make sure that running run.py successfully generates the result.txt file!

Submission

- Due: 25.06.11 23:59
- Submit file
 - Report (studentid_name.pdf)
 - Source code and state_dict(model.pth)
 - Must submit in student_id.zip / ex) 2023111111.zip
 - **Size of state_dict \leq 50MB**
- First submit
 - Due: 25.06.01 23:59 (not necessary)
 - Pre-score will be released on 06/03.

Grading

- Grade: 100 points total
- Report: 70 points
 - Evaluation will focus on how effectively and diversely you applied the concepts learned in class.
 - Report Format (maximum 4 pages):
 - Data
 - Model
 - Training Setting (include brief training code)
 - Result
- Performance: 30 points
 - Points: $30.3 - 0.3 * (\text{your rank})$
- The run.py file must only perform inference, not training or Non-runnable code → huge minus!
- Feel free to email TA if the bug isn't fixed or if you have any questions!
 - cjy9100@hanyang.ac.kr
 - suhoij47@hanyang.ac.kr