

NetworkVisualization-PyTorch

November 30, 2024

```
[1]: #COMMENT IF NOT USING COLAB VM

# This mounts your Google Drive to the Colab VM.
from google.colab import drive
drive.mount('/content/drive')

# TODO: Enter the foldername in your Drive where you have saved the unzipped
# assignment folder, e.g. 'CS6353/assignment5/'
FOLDERNAME = 'CS6353/assignment5/'
assert FOLDERNAME is not None, "[!] Enter the foldername."

# Now that we've mounted your Drive, this ensures that
# the Python interpreter of the Colab VM can load
# python files from within it.
import sys
sys.path.append('/content/drive/My Drive/{}'.format(FOLDERNAME))

# This downloads the CIFAR-10 dataset to your Drive
# if it doesn't already exist.
%cd /content/drive/My\ Drive/$FOLDERNAME/cs6353/datasets/
!bash get_datasets.sh
%cd /content/drive/My\ Drive/$FOLDERNAME
```

```
Mounted at /content/drive
/content/drive/My Drive/CS6353/assignment5/cs6353/datasets
--2024-11-30 18:36:39--
http://supermoe.cs.umass.edu/682/asgns/coco_captioning.zip
Resolving supermoe.cs.umass.edu (supermoe.cs.umass.edu)... 128.119.244.95
Connecting to supermoe.cs.umass.edu
(supermoe.cs.umass.edu)|128.119.244.95|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1035210391 (987M) [application/zip]
Saving to: 'coco_captioning.zip'

coco_captioning.zip 100%[=====>] 987.25M  6.94MB/s    in 6m 0s

2024-11-30 18:42:39 (2.74 MB/s) - 'coco_captioning.zip' saved
[1035210391/1035210391]
```

```

Archive:  coco_captioning.zip
replace coco_captioning/coco2014_captions.h5? [y]es, [n]o, [A]ll, [N]one,
[r]ename: Y
    inflating: coco_captioning/coco2014_captions.h5
replace coco_captioning/coco2014_vocab.json? [y]es, [n]o, [A]ll, [N]one,
[r]ename: Y
    inflating: coco_captioning/coco2014_vocab.json
replace coco_captioning/train2014_images.txt? [y]es, [n]o, [A]ll, [N]one,
[r]ename: Y
    inflating: coco_captioning/train2014_images.txt
replace coco_captioning/train2014_urls.txt? [y]es, [n]o, [A]ll, [N]one,
[r]ename: Y
    inflating: coco_captioning/train2014_urls.txt
replace coco_captioning/train2014_vgg16_fc7.h5? [y]es, [n]o, [A]ll, [N]one,
[r]ename: Y
    inflating: coco_captioning/train2014_vgg16_fc7.h5
replace coco_captioning/train2014_vgg16_fc7_pca.h5? [y]es, [n]o, [A]ll, [N]one,
[r]ename: Y
    inflating: coco_captioning/train2014_vgg16_fc7_pca.h5
replace coco_captioning/val2014_images.txt? [y]es, [n]o, [A]ll, [N]one,
[r]ename: Y
    inflating: coco_captioning/val2014_images.txt
replace coco_captioning/val2014_urls.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename:
Y
    inflating: coco_captioning/val2014_urls.txt
replace coco_captioning/val2014_vgg16_fc7.h5? [y]es, [n]o, [A]ll, [N]one,
[r]ename: Y
    inflating: coco_captioning/val2014_vgg16_fc7.h5
replace coco_captioning/val2014_vgg16_fc7_pca.h5? [y]es, [n]o, [A]ll, [N]one,
[r]ename: Y
    inflating: coco_captioning/val2014_vgg16_fc7_pca.h5
--2024-11-30 18:44:32--
http://supermoe.cs.umass.edu/682/asgns/squeezenet_tf.zip
Resolving supermoe.cs.umass.edu (supermoe.cs.umass.edu)... 128.119.244.95
Connecting to supermoe.cs.umass.edu
(supermoe.cs.umass.edu)|128.119.244.95|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9202140 (8.8M) [application/zip]
Saving to: 'squeezenet_tf.zip'

squeezenet_tf.zip  100%[=====>]    8.78M  1.63MB/s    in 10s

2024-11-30 18:44:42 (895 KB/s) - 'squeezenet_tf.zip' saved [9202140/9202140]

Archive:  squeezenet_tf.zip
replace squeezenet.ckpt.data-00000-of-00001? [y]es, [n]o, [A]ll, [N]one,
[r]ename: Y

```

```

inflating: squeezenet.ckpt.data-00000-of-00001
replace squeezenet.ckpt.index? [y]es, [n]o, [A]ll, [N]one, [r]ename: Y
inflating: squeezenet.ckpt.index
replace squeezenet.ckpt.meta? [y]es, [n]o, [A]ll, [N]one, [r]ename: Y
inflating: squeezenet.ckpt.meta
--2024-11-30 18:44:54--
http://supermoe.cs.umass.edu/682/asgns/imagenet_val_25.npz
Resolving supermoe.cs.umass.edu (supermoe.cs.umass.edu)... 128.119.244.95
Connecting to supermoe.cs.umass.edu
(supermoe.cs.umass.edu)|128.119.244.95|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3940548 (3.8M)
Saving to: 'imagenet_val_25.npz.1'

imagenet_val_25.npz 100%[=====>] 3.76M 2.00MB/s in 1.9s

2024-11-30 18:44:56 (2.00 MB/s) - 'imagenet_val_25.npz.1' saved
[3940548/3940548]

```

/content/drive/My Drive/CS6353/assignment5

```

[ ]: # #UNCOMMENT IF USING CADE
# import os
# ##### Request a GPU #####
# ## This function locates an available gpu for usage. In addition, this
# ↪function reserves a specified
# ## memory space exclusively for your account. The memory reservation prevents
# ↪the decrement in computational
# ## speed when other users try to allocate memory on the same gpu in the
# ↪shared systems, i.e., CADE machines.
# ## Note: If you use your own system which has a GPU with less than 4GB of
# ↪memory, remember to change the
# ## specified minimum memory.
# def define_gpu_to_use(minimum_memory_mb = 3500):
#     thres_memory = 600 #
#     gpu_to_use = None
#     try:
#         os.environ['CUDA_VISIBLE_DEVICES']
#         print('GPU already assigned before: ' + str(os.
# ↪environ['CUDA_VISIBLE_DEVICES']))
#         return
#     except:
#         pass

#     for i in range(16):
#         free_memory = !nvidia-smi --query-gpu=memory.free -i $i
# ↪--format=csv,nounits,noheader

```

```

#         if free_memory[0] == 'No devices were found':
#             break
#         free_memory = int(free_memory[0])

#         if free_memory > minimum_memory_mb - thres_memory:
#             gpu_to_use = i
#             break

#     if gpu_to_use is None:
#         print('Could not find any GPU available with the required free memory_
↳ of ' + str(minimum_memory_mb) \
#             + 'MB. Please use a different system for this assignment.')
#     else:
#         os.environ['CUDA_VISIBLE_DEVICES'] = str(gpu_to_use)
#         print('Chosen GPU: ' + str(gpu_to_use))

# ## Request a gpu and reserve the memory space
# define_gpu_to_use(4000)

```

1 Network Visualization (PyTorch)

In this notebook we will explore the use of *image gradients* for generating new images.

When training a model, we define a loss function which measures our current unhappiness with the model's performance; we then use backpropagation to compute the gradient of the loss with respect to the model parameters, and perform gradient descent on the model parameters to minimize the loss.

Here we will do something slightly different. We will start from a convolutional neural network model which has been pretrained to perform image classification on the ImageNet dataset. We will use this model to define a loss function which quantifies our current unhappiness with our image, then use backpropagation to compute the gradient of this loss with respect to the pixels of the image. We will then keep the model fixed, and perform gradient descent *on the image* to synthesize a new image which minimizes the loss.

In this notebook we will explore three techniques for image generation:

1. **Saliency Maps:** Saliency maps are a quick way to tell which part of the image influenced the classification decision made by the network.
2. **Fooling Images:** We can perturb an input image so that it appears the same to humans, but will be misclassified by the pretrained network.
3. **Class Visualization:** We can synthesize an image to maximize the classification score of a particular class; this can give us some sense of what the network is looking for when it classifies images of that class.

This notebook uses **PyTorch**;

```

[2]: import torch
import torchvision

```

```

import torchvision.transforms as T
import random
import numpy as np
from scipy.ndimage import gaussian_filter1d
import matplotlib.pyplot as plt
from cs6353.image_utils import SQUEEZENET_MEAN, SQUEEZENET_STD
from PIL import Image

%matplotlib inline
plt.rcParams['figure.figsize'] = (10.0, 8.0) # set default size of plots
plt.rcParams['image.interpolation'] = 'nearest'
plt.rcParams['image.cmap'] = 'gray'

```

1.0.1 Helper Functions

Our pretrained model was trained on images that had been preprocessed by subtracting the per-color mean and dividing by the per-color standard deviation. We define a few helper functions for performing and undoing this preprocessing. You don't need to do anything in this cell.

```

[3]: def preprocess(img, size=224):
    transform = T.Compose([
        T.Resize(size),
        T.ToTensor(),
        T.Normalize(mean=SQUEEZENET_MEAN.tolist(),
                    std=SQUEEZENET_STD.tolist()),
        T.Lambda(lambda x: x[None]),
    ])
    return transform(img)

def deprocess(img, should_rescale=True):
    transform = T.Compose([
        T.Lambda(lambda x: x[0]),
        T.Normalize(mean=[0, 0, 0], std=(1.0 / SQUEEZENET_STD).tolist()),
        T.Normalize(mean=(-SQUEEZENET_MEAN).tolist(), std=[1, 1, 1]),
        T.Lambda(rescale) if should_rescale else T.Lambda(lambda x: x),
        T.ToPILImage(),
    ])
    return transform(img)

def rescale(x):
    low, high = x.min(), x.max()
    x_rescaled = (x - low) / (high - low)
    return x_rescaled

def blur_image(X, sigma=1):
    X_np = X.cpu().clone().numpy()
    X_np = gaussian_filter1d(X_np, sigma, axis=2)

```

```

X_np = gaussian_filter1d(X_np, sigma, axis=3)
X.copy_(torch.Tensor(X_np).type_as(X))
return X

```

2 Pretrained Model

For all of our image generation experiments, we will start with a convolutional neural network which was pretrained to perform image classification on ImageNet. We can use any model here, but for the purposes of this assignment we will use SqueezeNet [1], which achieves accuracies comparable to AlexNet but with a significantly reduced parameter count and computational complexity.

Using SqueezeNet rather than AlexNet or VGG or ResNet means that we can easily perform all image generation experiments on CPU.

[1] Iandola et al, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5MB model size”, arXiv 2016

```

[4]: # Download and load the pretrained SqueezeNet model.
model = torchvision.models.squeezenet1_1(pretrained=True)

# We don't want to train the model, so tell PyTorch not to compute gradients
# with respect to model parameters.
for param in model.parameters():
    param.requires_grad = False

# you may see warning regarding initialization deprecated, that's fine, please
↳ continue to next steps

```

```

/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:208:
UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be
removed in the future, please use 'weights' instead.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:223:
UserWarning: Arguments other than a weight enum or `None` for 'weights' are
deprecated since 0.13 and may be removed in the future. The current behavior is
equivalent to passing `weights=SqueezeNet1_1_Weights.IMAGENET1K_V1`. You can
also use `weights=SqueezeNet1_1_Weights.DEFAULT` to get the most up-to-date
weights.
  warnings.warn(msg)
Downloading: "https://download.pytorch.org/models/squeezenet1_1-b8a52dc0.pth" to
/root/.cache/torch/hub/checkpoints/squeezenet1_1-b8a52dc0.pth
100%|          | 4.73M/4.73M [00:00<00:00, 21.0MB/s]

```

2.1 Load some ImageNet images

We have provided a few example images from the validation set of the ImageNet ILSVRC 2012 Classification dataset. To download these images, descend into `cs6353/datasets/` and run `get_imagenet_val.sh`.

Since they come from the validation set, our pretrained model did not see these images during training.

Run the following cell to visualize some of these images, along with their ground-truth labels.

```
[5]: from cs6353.data_utils import load_imagenet_val
X, y, class_names = load_imagenet_val(num=5)

plt.figure(figsize=(12, 6))
for i in range(5):
    plt.subplot(1, 5, i + 1)
    plt.imshow(X[i])
    plt.title(class_names[y[i]])
    plt.axis('off')
plt.gcf().tight_layout()
```



3 Saliency Maps

Using this pretrained model, we will compute class saliency maps as described in Section 3.1 of [2].

A **saliency map** tells us the degree to which each pixel in the image affects the classification score for that image. To compute it, we compute the gradient of the unnormalized score corresponding to the correct class (which is a scalar) with respect to the pixels of the image. If the image has shape $(3, H, W)$ then this gradient will also have shape $(3, H, W)$; for each pixel in the image, this gradient tells us the amount by which the classification score will change if the pixel changes by a small amount. To compute the saliency map, we take the absolute value of this gradient, then take the maximum value over the 3 input channels; the final saliency map thus has shape (H, W) and all entries are nonnegative.

[2] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, ICLR Workshop 2014.

3.0.1 Hint: PyTorch gather method

Recall in Assignment 1 you needed to select one element from each row of a matrix; if \mathbf{s} is a numpy array of shape (N, C) and \mathbf{y} is a numpy array of shape $(N,)$ containing integers $0 \leq y[i] < C$, then $\mathbf{s}[\text{np.arange}(N), \mathbf{y}]$ is a numpy array of shape $(N,)$ which selects one element from each element in \mathbf{s} using the indices in \mathbf{y} .

In PyTorch you can perform the same operation using the `gather()` method. If `s` is a PyTorch Tensor of shape `(N, C)` and `y` is a PyTorch Tensor of shape `(N,)` containing longs in the range $0 \leq y[i] < C$, then

```
s.gather(1, y.view(-1, 1)).squeeze()
```

will be a PyTorch Tensor of shape `(N,)` containing one entry from each row of `s`, selected according to the indices in `y`.

run the following cell to see an example.

You can also read the documentation for [the gather method](#) and [the squeeze method](#).

```
[6]: # Example of using gather to select one entry from each row in PyTorch
def gather_example():
    N, C = 4, 5
    s = torch.randn(N, C)
    y = torch.LongTensor([1, 2, 1, 3])
    print(s)
    print(y)
    print(s.gather(1, y.view(-1, 1)).squeeze())
gather_example()
```

```
tensor([[ 0.6616, -1.7859, -0.3630, -0.9565, -0.0309],
        [ 1.0160, -0.4174,  0.6025,  1.6517,  0.2762],
        [-1.0798, -0.5526,  1.0245, -1.3062, -1.4973],
        [ 1.0976,  2.7126,  0.5053,  0.7464,  0.0660]])
tensor([1, 2, 1, 3])
tensor([-1.7859,  0.6025, -0.5526,  0.7464])
```

```
[10]: def compute_saliency_maps(X, y, model):
    """
    Compute a class saliency map using the model for images X and labels y.

    Input:
    - X: Input images; Tensor of shape (N, 3, H, W)
    - y: Labels for X; LongTensor of shape (N,)
    - model: A pretrained CNN that will be used to compute the saliency map.

    Returns:
    - saliency: A Tensor of shape (N, H, W) giving the saliency maps for the
    ↪ input
    images.
    """
    # Make sure the model is in "test" mode
    model.eval()

    # Make input tensor require gradient
    X.requires_grad_()
```



```

# Perform a forward pass through the model
scores = model(X) # Shape: (N, C), where C is the number of classes

# Select the correct class scores using the labels y
correct_class_scores = scores.gather(1, y.view(-1, 1)).squeeze() # Shape:
↪(N,)

# Compute the gradient of the correct class scores with respect to the
↪input images
correct_class_scores.sum().backward()

# Compute the absolute value of the gradients and take the max across the
↪color channels
saliency = X.grad.abs().max(dim=1)[0] # Shape: (N, H, W)

return saliency

```

Once you have completed the implementation in the cell above, run the following to visualize some class saliency maps on our example images from the ImageNet validation set:

```

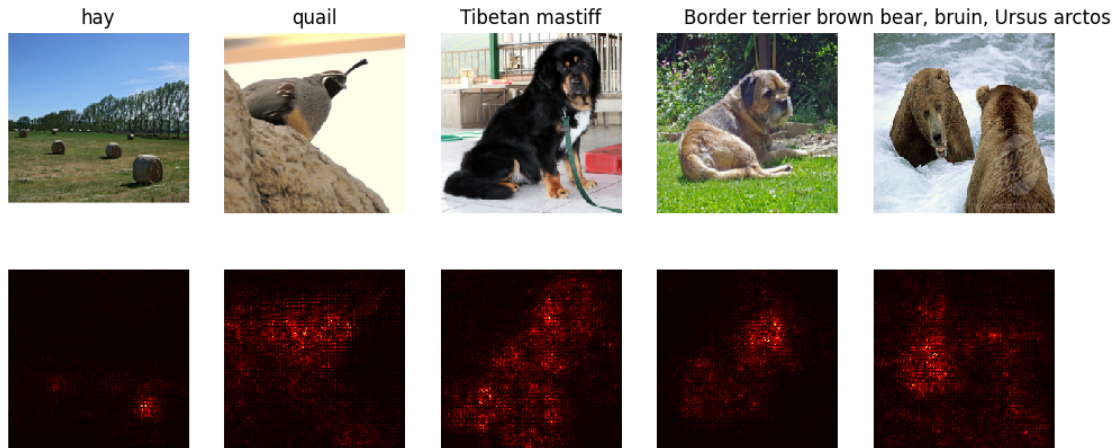
[11]: def show_saliency_maps(X, y):
# Convert X and y from numpy arrays to Torch Tensors
X_tensor = torch.cat([preprocess(Image.fromarray(x)) for x in X], dim=0)
y_tensor = torch.LongTensor(y)

# Compute saliency maps for images in X
saliency = compute_saliency_maps(X_tensor, y_tensor, model)

# Convert the saliency map from Torch Tensor to numpy array and show images
# and saliency maps together.
saliency = saliency.numpy()
N = X.shape[0]
for i in range(N):
    plt.subplot(2, N, i + 1)
    plt.imshow(X[i])
    plt.axis('off')
    plt.title(class_names[y[i]])
    plt.subplot(2, N, N + i + 1)
    plt.imshow(saliency[i], cmap=plt.cm.hot)
    plt.axis('off')
    plt.gcf().set_size_inches(12, 5)
plt.show()

show_saliency_maps(X, y)

```



4 INLINE QUESTION

A friend of yours suggests that in order to find an image that maximizes the correct score, we can perform gradient ascent on the input image, but instead of the gradient we can actually use the saliency map in each step to update the image. Is this assertion true? Why or why not?

ANS: The assertion is false because the saliency map is not the same as the gradient. While the gradient provides directionality for optimizing the image to maximize the correct score, the saliency map only shows the magnitude of importance for each region, losing directional information by taking the absolute value and reducing across channels. Using the saliency map for gradient ascent would fail to make meaningful updates, as it lacks the necessary directional cues to increase the score effectively. Thus, the gradient, not the saliency map, should be used for optimization.

5 Fooling Images

We can also use image gradients to generate “fooling images” as discussed in [3]. Given an image and a target class, we can perform gradient **ascent** over the image to maximize the target class, stopping when the network classifies the image as the target class. Implement the following function to generate fooling images.

[3] Szegedy et al, “Intriguing properties of neural networks”, ICLR 2014

```
[12]: def make_fooling_image(X, target_y, model):
    """
    Generate a fooling image that is close to X, but that the model classifies
    as target_y.

    Inputs:
    - X: Input image; Tensor of shape (1, 3, 224, 224)
    - target_y: An integer in the range [0, 1000)
    - model: A pretrained CNN
```

```

Returns:
- X_fooling: An image that is close to X, but that is classified as target_y
  by the model.
"""
# Initialize our fooling image to the input image, and make it require_
↪gradient
X_fooling = X.clone()
X_fooling = X_fooling.requires_grad_()

learning_rate = 1

for i in range(100): # Perform up to 100 iterations of gradient ascent
    # Set model to evaluation mode
    model.eval()

    # Forward pass to compute the scores
    scores = model(X_fooling) # Shape: (1, 1000)

    # Check if the model is fooled
    ↪if scores.argmax(dim=1).item() == target_y:
        print(f"Model fooled after {i+1} iterations.")
        break

    # Compute the gradient of the target class score with respect to the_
↪image
    target_score = scores[0, target_y]
    model.zero_grad()
    target_score.backward()

    # Normalize the gradient
    grad = X_fooling.grad.data
    grad_norm = grad.norm() + 1e-8 # Avoid division by zero
    dX = learning_rate * grad / grad_norm

    # Update the fooling image
    X_fooling.data += dX

    # Zero out the gradient for the next iteration
    X_fooling.grad.data.zero_()

return X_fooling

```

Run the following cell to generate a fooling image. You should ideally see at first glance no major difference between the original and fooling images, and the network should now make an incorrect prediction on the fooling one. However you should see a bit of random noise if you look at the 10x magnified difference between the original and fooling images. Feel free to change the `idx` variable

to explore other images.

```
[13]: idx = 0
      target_y = 6

      X_tensor = torch.cat([preprocess(Image.fromarray(x)) for x in X], dim=0)
      X_fooling = make_fooling_image(X_tensor[idx:idx+1], target_y, model)

      scores = model(X_fooling)
      assert target_y == scores.data.max(1)[1][0].item(), 'The model is not fooled!'
```

Model fooled after 10 iterations.

After generating a fooling image, run the following cell to visualize the original image, the fooling image, as well as the difference between them.

```
[14]: X_fooling_np = deprocess(X_fooling.clone())
      X_fooling_np = np.asarray(X_fooling_np).astype(np.uint8)

      plt.subplot(1, 4, 1)
      plt.imshow(X[idx])
      plt.title(class_names[y[idx]])
      plt.axis('off')

      plt.subplot(1, 4, 2)
      plt.imshow(X_fooling_np)
      plt.title(class_names[target_y])
      plt.axis('off')

      plt.subplot(1, 4, 3)
      X_pre = preprocess(Image.fromarray(X[idx]))
      diff = np.asarray(deprocess(X_fooling - X_pre, should_rescale=False))
      plt.imshow(diff)
      plt.title('Difference')
      plt.axis('off')

      plt.subplot(1, 4, 4)
      diff = np.asarray(deprocess(10 * (X_fooling - X_pre), should_rescale=False))
      plt.imshow(diff)
      plt.title('Magnified difference (10x)')
      plt.axis('off')

      plt.gcf().set_size_inches(12, 5)
      plt.show()
```



6 Class visualization

By starting with a random noise image and performing gradient ascent on a target class, we can generate an image that the network will recognize as the target class. This idea was first presented in [2]; [3] extended this idea by suggesting several regularization techniques that can improve the quality of the generated image.

Concretely, let I be an image and let y be a target class. Let $s_y(I)$ be the score that a convolutional network assigns to the image I for class y ; note that these are raw unnormalized scores, not class probabilities. We wish to generate an image I^* that achieves a high score for the class y by solving the problem

$$I^* = \arg \max_I (s_y(I) - R(I))$$

where R is a (possibly implicit) regularizer (note the sign of $R(I)$ in the argmax: we want to minimize this regularization term). We can solve this optimization problem using gradient ascent, computing gradients with respect to the generated image. We will use (explicit) L2 regularization of the form

$$R(I) = \lambda \|I\|_2^2$$

and implicit regularization as suggested by [3] by periodically blurring the generated image. We can solve this problem using gradient ascent on the generated image.

In the cell below, complete the implementation of the `create_class_visualization` function.

[2] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, ICLR Workshop 2014.

[3] Yosinski et al, “Understanding Neural Networks Through Deep Visualization”, ICML 2015 Deep Learning Workshop

```
[15]: def jitter(X, ox, oy):
      """
      Helper function to randomly jitter an image.

      Inputs
```

- *X*: PyTorch Tensor of shape (N, C, H, W)
 - *ox, oy*: Integers giving number of pixels to jitter along W and H axes

Returns: A new PyTorch Tensor of shape (N, C, H, W)

```
"""
if ox != 0:
    left = X[:, :, :, :-ox]
    right = X[:, :, :, -ox:]
    X = torch.cat([right, left], dim=3)
if oy != 0:
    top = X[:, :, :-oy]
    bottom = X[:, :, -oy:]
    X = torch.cat([bottom, top], dim=2)
return X
```

```
[16]: def create_class_visualization(target_y, model, dtype, **kwargs):
    """
    Generate an image to maximize the score of target_y under a pretrained_
    ↪ model.

    Inputs:
    - target_y: Integer in the range [0, 1000) giving the index of the class
    - model: A pretrained CNN that will be used to generate the image
    - dtype: Torch datatype to use for computations

    Keyword arguments:
    - l2_reg: Strength of L2 regularization on the image
    - learning_rate: How big of a step to take
    - num_iterations: How many iterations to use
    - blur_every: How often to blur the image as an implicit regularizer
    - max_jitter: How much to jitter the image as an implicit regularizer
    - show_every: How often to show the intermediate result
    """
    model.type(dtype)
    l2_reg = kwargs.pop('l2_reg', 1e-3)
    learning_rate = kwargs.pop('learning_rate', 25)
    num_iterations = kwargs.pop('num_iterations', 100)
    blur_every = kwargs.pop('blur_every', 10)
    max_jitter = kwargs.pop('max_jitter', 16)
    show_every = kwargs.pop('show_every', 25)

    # Randomly initialize the image as a PyTorch Tensor, and make it require_
    ↪ gradient.
    img = torch.randn(1, 3, 224, 224).mul_(1.0).type(dtype).requires_grad_()

    for t in range(num_iterations):
        # Randomly jitter the image a bit; this gives slightly nicer results
```

```

ox, oy = random.randint(0, max_jitter), random.randint(0, max_jitter)
img.data.copy_(jitter(img.data, ox, oy))

#####
# Use the model to compute the gradient of the score for the      #
# class target_y with respect to the pixels of the image, and make a  #
# gradient step on the image using the learning rate. Include the L2  #
# regularization term to prevent overfitting to specific pixels.      #
#####

# Forward pass to compute scores
scores = model(img)

# Extract the score for the target class
target_score = scores[0, target_y]

# Compute the L2 regularization term
l2_loss = l2_reg * (img**2).sum()

# Combine the target score and L2 regularization term
loss = target_score - l2_loss

# Zero out gradients from the previous iteration
model.zero_grad()

# Backward pass to compute gradients
loss.backward()

# Normalize the gradient and take a gradient ascent step
grad = img.grad.data
grad_norm = grad.norm() + 1e-8 # Prevent division by zero
img.data += learning_rate * grad / grad_norm

# Clear gradients for the next iteration
img.grad.data.zero_()

#####
#                                     END OF YOUR CODE                                     #
#####

# Undo the random jitter
img.data.copy_(jitter(img.data, -ox, -oy))

# As regularizer, clamp and periodically blur the image
for c in range(3):
    lo = float(-SQUEEZENET_MEAN[c] / SQUEEZENET_STD[c])
    hi = float((1.0 - SQUEEZENET_MEAN[c]) / SQUEEZENET_STD[c])

```

```

        img.data[:, c].clamp_(min=lo, max=hi)
    if t % blur_every == 0:
        blur_image(img.data, sigma=0.5)

    # Periodically show the image
    if t == 0 or (t + 1) % show_every == 0 or t == num_iterations - 1:
        plt.imshow(deprocess(img.data.clone().cpu()))
        class_name = class_names[target_y]
        plt.title('%s\nIteration %d / %d' % (class_name, t + 1,
↪num_iterations))
        plt.gcf().set_size_inches(4, 4)
        plt.axis('off')
        plt.show()

    return deprocess(img.data.cpu())

```

Once you have completed the implementation in the cell above, run the following cell to generate an image of a Tarantula:

```

[17]: dtype = torch.FloatTensor
      # dtype = torch.cuda.FloatTensor # Uncomment this to use GPU
      model.type(dtype)

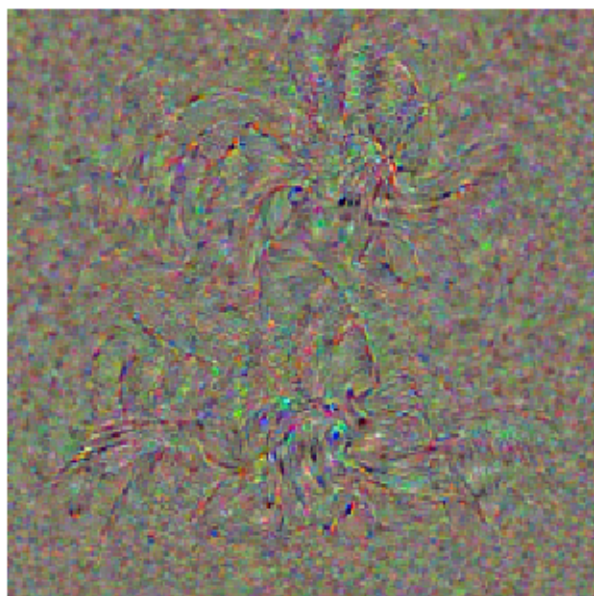
      target_y = 76 # Tarantula
      # target_y = 78 # Tick
      # target_y = 187 # Yorkshire Terrier
      # target_y = 683 # Oboe
      # target_y = 366 # Gorilla
      # target_y = 604 # Hourglass
      out = create_class_visualization(target_y, model, dtype)

```

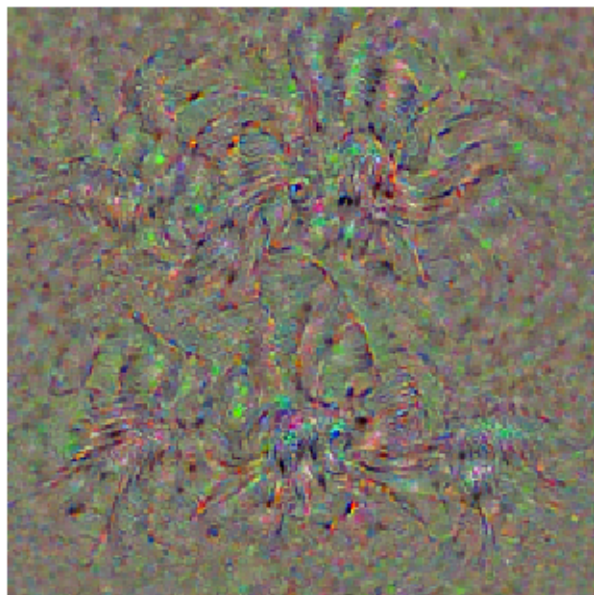

tarantula
Iteration 1 / 100



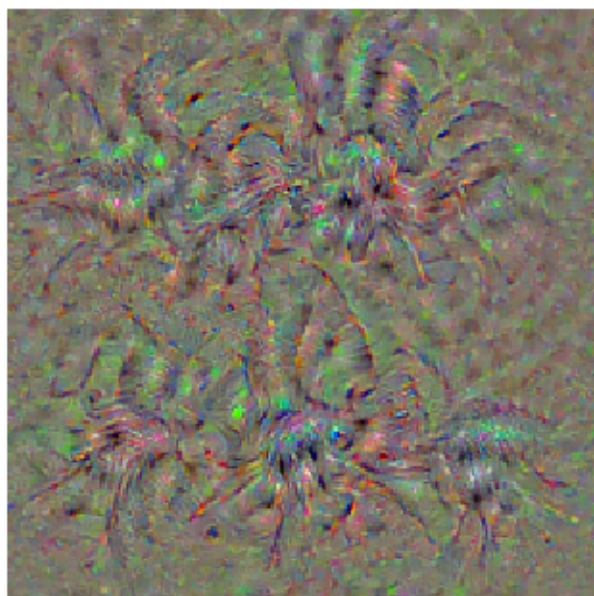
tarantula
Iteration 25 / 100



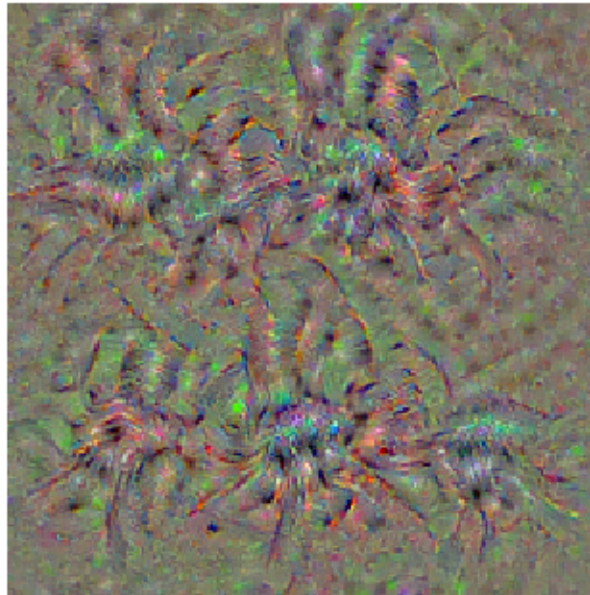
tarantula
Iteration 50 / 100



tarantula
Iteration 75 / 100



tarantula
Iteration 100 / 100

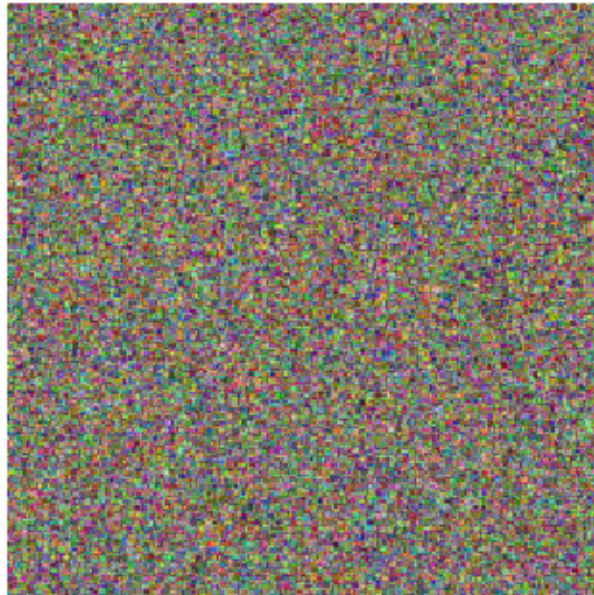


Try out your class visualization on other classes! You should also feel free to play with various hyperparameters to try and improve the quality of the generated image, but this is not required.

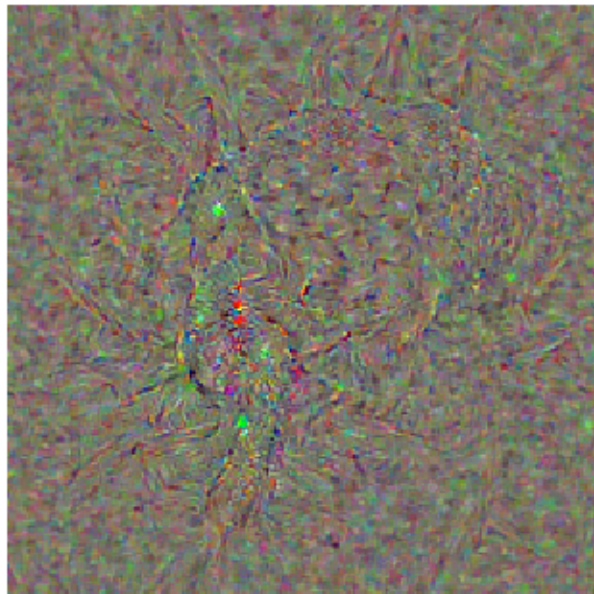
```
[18]: # target_y = 78 # Tick
target_y = 366 # Yorkshire Terrier
# target_y = 683 # Oboe
# target_y = 366 # Gorilla
# target_y = 604 # Hourglass
target_y = np.random.randint(1000)
print(class_names[target_y])
X = create_class_visualization(target_y, model, dtype)
```

stinkhorn, carrion fungus

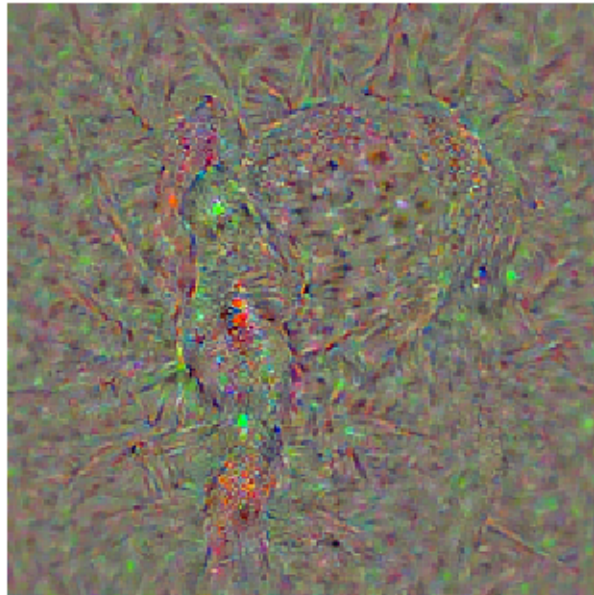
stinkhorn, carrion fungus
Iteration 1 / 100



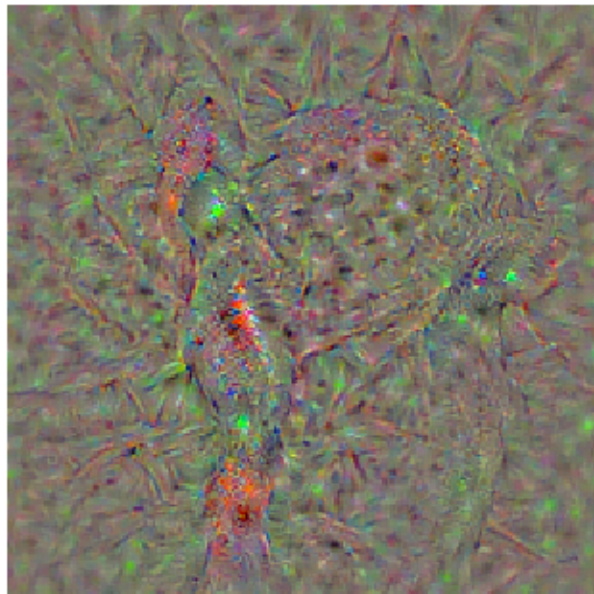
stinkhorn, carrion fungus
Iteration 25 / 100



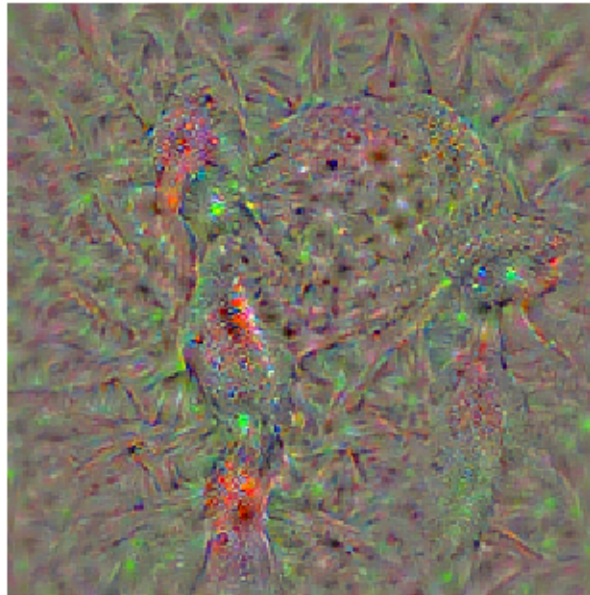
stinkhorn, carrion fungus
Iteration 50 / 100



stinkhorn, carrion fungus
Iteration 75 / 100



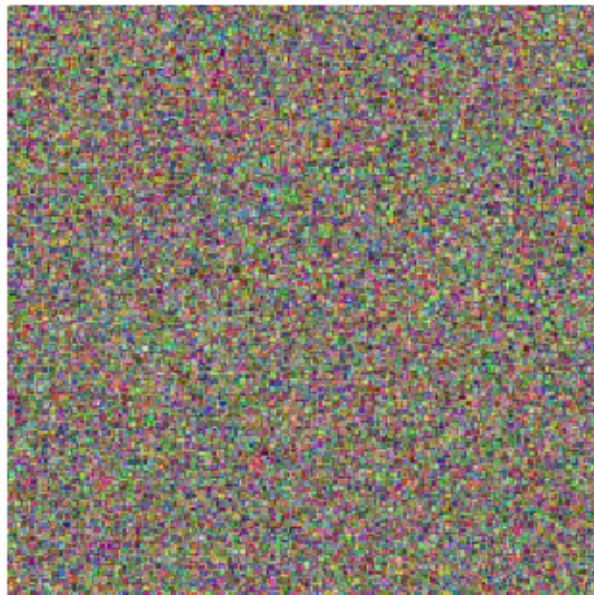
stinkhorn, carrion fungus
Iteration 100 / 100



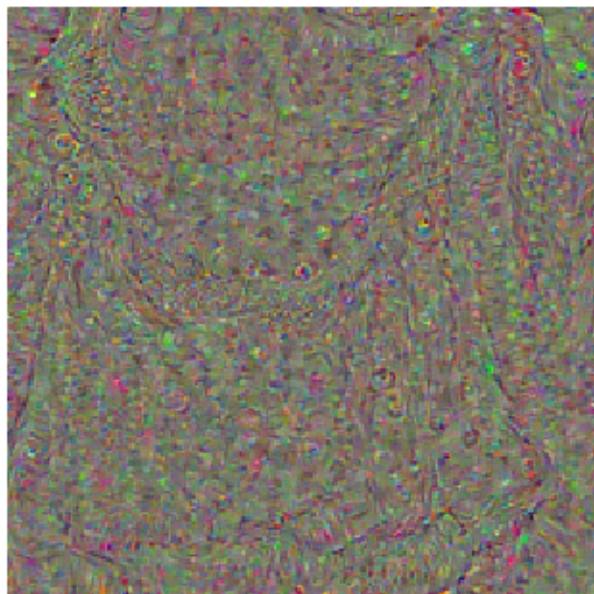
```
[19]: # target_y = 78 # Tick
target_y = 366 # Yorkshire Terrier
# target_y = 683 # Oboe
# target_y = 366 # Gorilla
# target_y = 604 # Hourglass
target_y = np.random.randint(1000)
print(class_names[target_y])
X = create_class_visualization(target_y, model, dtype)
```

overskirt

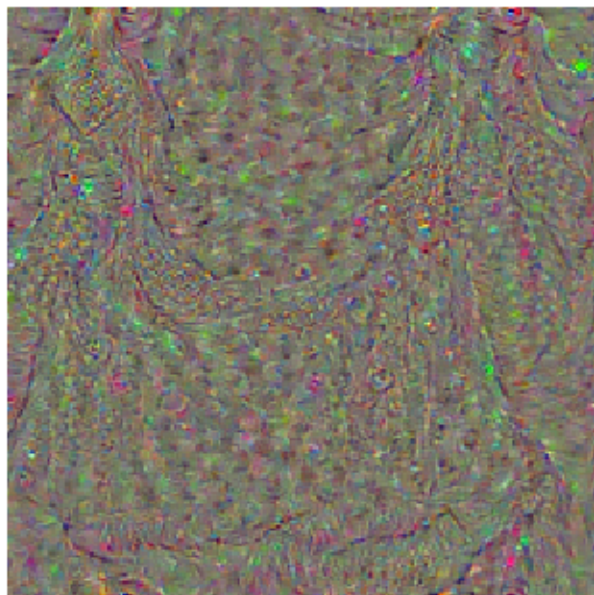
overskirt
Iteration 1 / 100



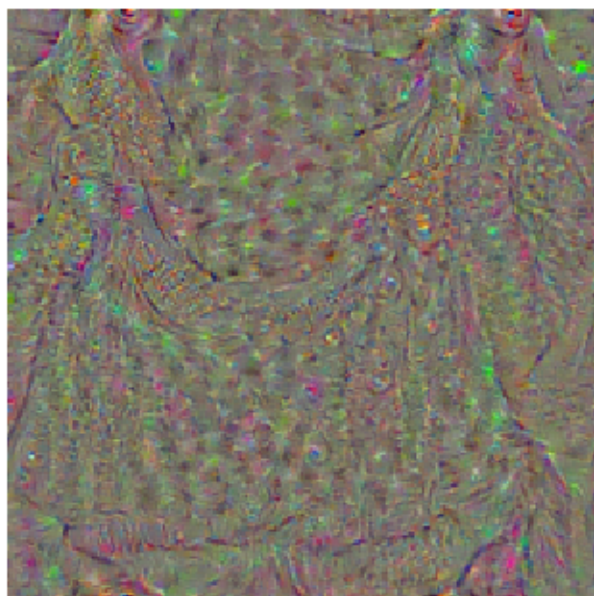
overskirt
Iteration 25 / 100



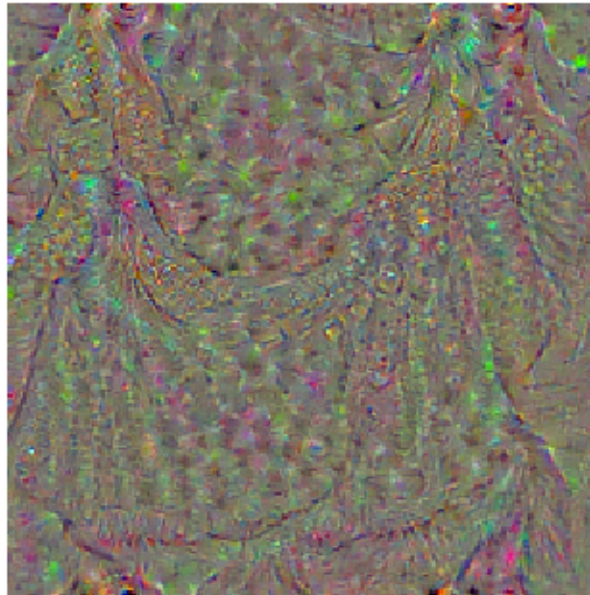
overskirt
Iteration 50 / 100



overskirt
Iteration 75 / 100



overskirt
Iteration 100 / 100



```
[ ]: ! sudo apt-get install texlive-xetex texlive-fonts-recommended_
↳texlive-plain-generic pandoc
```

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

The following additional packages will be installed:

dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono
fonts-texgyre fonts-urw-base35 libapache-pom-java
libcmark-gfm-extensions0.29.0.gfm.3 libcmark-gfm0.29.0.gfm.3
libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common libidn12 libijs-0.35 libjbig2dec0 libkpathsea6
libpdfbox-java libptexenc1 libruby3.0 libsynchronet2 libteckit0 libtexlua53
libtexluajit2 libwoff1 libzip-0-13 lmodern pandoc-data poppler-data
preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-webrick
ruby-xmlrpc ruby3.0 rubygems-integration t1utils teckit tex-common tex-gyre
texlive-base texlive-binaries texlive-latex-base texlive-latex-extra
texlive-latex-recommended texlive-pictures tipa xfonts-encodings
xfonts-utils

Suggested packages:

fonts-noto fonts-freefont-otf | fonts-freefont-ttf libavalon-framework-java
libcommons-logging-java-doc libexcalibur-logkit-java liblog4j1.2-java
texlive-luatex pandoc-citeproc context wkhtmltopdf librsvg2-bin groff ghc
nodejs php python libjs-mathjax libjs-katex citation-style-language-styles
poppler-utils ghostscript fonts-japanese-mincho | fonts-ipafont-mincho

```

fonts-japanese-gothic | fonts-ipafont-gothic fonts-arphic-ukai
fonts-arphic-uming fonts-nanum ri ruby-dev bundler debhelper gv
| postscript-viewer perl-tk xpdf | pdf-viewer xzdec
texlive-fonts-recommended-doc texlive-latex-base-doc python3-pygments
icc-profiles libfile-which-perl libspreadsheet-parseexcel-perl
texlive-latex-extra-doc texlive-latex-recommended-doc texlive-pstricks
dot2tex prerex texlive-pictures-doc vprerex default-jre-headless tipa-doc
The following NEW packages will be installed:
dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono
fonts-texgyre fonts-urw-base35 libapache-pom-java
libcmark-gfm-extensions0.29.0.gfm.3 libcmark-gfm0.29.0.gfm.3
libcommons-logging-java libcommons-parent-java libfontbox-java libfontenc1
libgs9 libgs9-common libidn12 libijs-0.35 libjbig2dec0 libkpathsea6
libpdfbox-java libptexenc1 libruby3.0 libsystex2 libteckit0 libtexlua53
libtexluajit2 libwoff1 libzip-0-13 lmodern pandoc pandoc-data poppler-data
preview-latex-style rake ruby ruby-net-telnet ruby-rubygems ruby-webrick
ruby-xmllrpc ruby3.0 rubygems-integration t1utils teckit tex-common tex-gyre
texlive-base texlive-binaries texlive-fonts-recommended texlive-latex-base
texlive-latex-extra texlive-latex-recommended texlive-pictures
texlive-plain-generic texlive-xetex tipa xfonts-encodings xfonts-utils
0 upgraded, 58 newly installed, 0 to remove and 49 not upgraded.
Need to get 202 MB of archives.
After this operation, 728 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-droid-fallback all
1:6.0.1r16-1.1build1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-lato all 2.0-2.1
[2,696 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 poppler-data all
0.4.11-1 [2,171 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-common all 6.17
[33.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-urw-base35 all
20200910-1 [6,367 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9-common
all 9.55.0~dfsg1-0ubuntu5.10 [752 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libidn12 amd64
1.38-4ubuntu1 [60.0 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 libijs-0.35 amd64
0.35-15build2 [16.5 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 libjbig2dec0 amd64
0.19-3build2 [64.7 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9 amd64
9.55.0~dfsg1-0ubuntu5.10 [5,031 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libkpathsea6
amd64 2021.20210626.59705-1ubuntu0.2 [60.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 libwoff1 amd64
1.0.2-1build4 [45.2 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy/universe amd64 dvisvgm amd64

```

2.13.1-1 [1,221 kB]
Get:14 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 fonts-lmodern all 2.004.5-6.1 [4,532 kB]
Get:15 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 fonts-noto-mono all 20201225-1build1 [397 kB]
Get:16 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 fonts-texgyre all 20180621-3.1 [10.2 MB]
Get:17 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libapache-pom-java all 18-1 [4,720 B]
Get:18 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libcmark-gfm0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [115 kB]
Get:19 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libcmark-gfm-extensions0.29.0.gfm.3 amd64 0.29.0.gfm.3-3 [25.1 kB]
Get:20 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libcommons-parent-java all 43-1 [10.8 kB]
Get:21 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libcommons-logging-java all 1.2-2 [60.3 kB]
Get:22 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 libfontenc1 amd64 1:1.1.4-1build3 [14.7 kB]
Get:23 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libptexenc1 amd64 2021.20210626.59705-1ubuntu0.2 [39.1 kB]
Get:24 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 rubygems-integration all 1.18 [5,336 B]
Get:25 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 ruby3.0 amd64 3.0.2-7ubuntu2.8 [50.1 kB]
Get:26 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 ruby-rubygems all 3.3.5-2 [228 kB]
Get:27 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 ruby amd64 1:3.0~exp1 [5,100 B]
Get:28 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 rake all 13.0.6-2 [61.7 kB]
Get:29 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 ruby-net-telnet all 0.1.1-2 [12.6 kB]
Get:30 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 ruby-webrick all 1.7.0-3ubuntu0.1 [52.1 kB]
Get:31 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 ruby-xmlrpc all 0.3.2-1ubuntu0.1 [24.9 kB]
Get:32 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libruby3.0 amd64 3.0.2-7ubuntu2.8 [5,113 kB]
Get:33 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libsynchronet2 amd64 2021.20210626.59705-1ubuntu0.2 [55.6 kB]
Get:34 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libteckit0 amd64 2.5.11+ds1-1 [421 kB]
Get:35 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libtexlua53 amd64 2021.20210626.59705-1ubuntu0.2 [120 kB]
Get:36 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libtexluajit2 amd64 2021.20210626.59705-1ubuntu0.2 [267 kB]
Get:37 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libzip-0-13 amd64

```

0.13.72+dfsg.1-1.1 [27.0 kB]
Get:38 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-encodings all
1:1.0.5-0ubuntu2 [578 kB]
Get:39 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-utils amd64
1:7.7+6build2 [94.6 kB]
Get:40 http://archive.ubuntu.com/ubuntu jammy/universe amd64 lmodern all
2.004.5-6.1 [9,471 kB]
Get:41 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pandoc-data all
2.9.2.1-3ubuntu2 [81.8 kB]
Get:42 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pandoc amd64
2.9.2.1-3ubuntu2 [20.3 MB]
Get:43 http://archive.ubuntu.com/ubuntu jammy/universe amd64 preview-latex-style
all 12.2-1ubuntu1 [185 kB]
Get:44 http://archive.ubuntu.com/ubuntu jammy/main amd64 t1utils amd64
1.41-4build2 [61.3 kB]
Get:45 http://archive.ubuntu.com/ubuntu jammy/universe amd64 teckit amd64
2.5.11+ds1-1 [699 kB]
Get:46 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-gyre all
20180621-3.1 [6,209 kB]
Get:47 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 texlive-
binaries amd64 2021.20210626.59705-1ubuntu0.2 [9,860 kB]
Get:48 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-base all
2021.20220204-1 [21.0 MB]
Get:49 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-fonts-
recommended all 2021.20220204-1 [4,972 kB]
Get:50 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-base
all 2021.20220204-1 [1,128 kB]
Get:51 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libfontbox-java all
1:1.8.16-2 [207 kB]
Get:52 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libpdfbox-java all
1:1.8.16-2 [5,199 kB]
Get:53 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-
recommended all 2021.20220204-1 [14.4 MB]
Get:54 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-pictures
all 2021.20220204-1 [8,720 kB]
Get:55 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-extra
all 2021.20220204-1 [13.9 MB]
Get:56 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-plain-
generic all 2021.20220204-1 [27.5 MB]
Get:57 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tipa all 2:1.3-21
[2,967 kB]
Get:58 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-xetex all
2021.20220204-1 [12.4 MB]
Fetched 202 MB in 6s (31.2 MB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78,
<> line 58.)

```

```

debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package fonts-droid-fallback.
(Reading database ... 123630 files and directories currently installed.)
Preparing to unpack .../00-fonts-droid-fallback_1%3a6.0.1r16-1.1build1_all.deb
...
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1build1) ...
Selecting previously unselected package fonts-lato.
Preparing to unpack .../01-fonts-lato_2.0-2.1_all.deb ...
Unpacking fonts-lato (2.0-2.1) ...
Selecting previously unselected package poppler-data.
Preparing to unpack .../02-poppler-data_0.4.11-1_all.deb ...
Unpacking poppler-data (0.4.11-1) ...
Selecting previously unselected package tex-common.
Preparing to unpack .../03-tex-common_6.17_all.deb ...
Unpacking tex-common (6.17) ...
Selecting previously unselected package fonts-urw-base35.
Preparing to unpack .../04-fonts-urw-base35_20200910-1_all.deb ...
Unpacking fonts-urw-base35 (20200910-1) ...
Selecting previously unselected package libgs9-common.
Preparing to unpack .../05-libgs9-common_9.55.0~dfsg1-0ubuntu5.10_all.deb ...
Unpacking libgs9-common (9.55.0~dfsg1-0ubuntu5.10) ...
Selecting previously unselected package libidn12:amd64.
Preparing to unpack .../06-libidn12_1.38-4ubuntu1_amd64.deb ...
Unpacking libidn12:amd64 (1.38-4ubuntu1) ...
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack .../07-libijs-0.35_0.35-15build2_amd64.deb ...
Unpacking libijs-0.35:amd64 (0.35-15build2) ...
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack .../08-libjbig2dec0_0.19-3build2_amd64.deb ...
Unpacking libjbig2dec0:amd64 (0.19-3build2) ...
Selecting previously unselected package libgs9:amd64.
Preparing to unpack .../09-libgs9_9.55.0~dfsg1-0ubuntu5.10_amd64.deb ...
Unpacking libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.10) ...
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack .../10-libkpathsea6_2021.20210626.59705-1ubuntu0.2_amd64.deb
...
Unpacking libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package libwoff1:amd64.
Preparing to unpack .../11-libwoff1_1.0.2-1build4_amd64.deb ...
Unpacking libwoff1:amd64 (1.0.2-1build4) ...
Selecting previously unselected package dvisvgm.
Preparing to unpack .../12-dvisvgm_2.13.1-1_amd64.deb ...
Unpacking dvisvgm (2.13.1-1) ...
Selecting previously unselected package fonts-lmodern.

```

```

Preparing to unpack .../13-fonts-lmodern_2.004.5-6.1_all.deb ...
Unpacking fonts-lmodern (2.004.5-6.1) ...
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack .../14-fonts-noto-mono_20201225-1build1_all.deb ...
Unpacking fonts-noto-mono (20201225-1build1) ...
Selecting previously unselected package fonts-texgyre.
Preparing to unpack .../15-fonts-texgyre_20180621-3.1_all.deb ...
Unpacking fonts-texgyre (20180621-3.1) ...
Selecting previously unselected package libapache-pom-java.
Preparing to unpack .../16-libapache-pom-java_18-1_all.deb ...
Unpacking libapache-pom-java (18-1) ...
Selecting previously unselected package libcmark-gfm0.29.0.gfm.3:amd64.
Preparing to unpack .../17-libcmark-gfm0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb ...
Unpacking libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) ...
Selecting previously unselected package libcmark-gfm-
extensions0.29.0.gfm.3:amd64.
Preparing to unpack .../18-libcmark-gfm-
extensions0.29.0.gfm.3_0.29.0.gfm.3-3_amd64.deb ...
Unpacking libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) ...
Selecting previously unselected package libcommons-parent-java.
Preparing to unpack .../19-libcommons-parent-java_43-1_all.deb ...
Unpacking libcommons-parent-java (43-1) ...
Selecting previously unselected package libcommons-logging-java.
Preparing to unpack .../20-libcommons-logging-java_1.2-2_all.deb ...
Unpacking libcommons-logging-java (1.2-2) ...
Selecting previously unselected package libfontenc1:amd64.
Preparing to unpack .../21-libfontenc1_1%3a1.1.4-1build3_amd64.deb ...
Unpacking libfontenc1:amd64 (1:1.1.4-1build3) ...
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack .../22-libptexenc1_2021.20210626.59705-1ubuntu0.2_amd64.deb
...
Unpacking libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package rubygems-integration.
Preparing to unpack .../23-rubygems-integration_1.18_all.deb ...
Unpacking rubygems-integration (1.18) ...
Selecting previously unselected package ruby3.0.
Preparing to unpack .../24-ruby3.0_3.0.2-7ubuntu2.8_amd64.deb ...
Unpacking ruby3.0 (3.0.2-7ubuntu2.8) ...
Selecting previously unselected package ruby-rubygems.
Preparing to unpack .../25-ruby-rubygems_3.3.5-2_all.deb ...
Unpacking ruby-rubygems (3.3.5-2) ...
Selecting previously unselected package ruby.
Preparing to unpack .../26-ruby_1%3a3.0~exp1_amd64.deb ...
Unpacking ruby (1:3.0~exp1) ...
Selecting previously unselected package rake.
Preparing to unpack .../27-rake_13.0.6-2_all.deb ...
Unpacking rake (13.0.6-2) ...
Selecting previously unselected package ruby-net-telnet.

```

```

Preparing to unpack .../28-ruby-net-telnet_0.1.1-2_all.deb ...
Unpacking ruby-net-telnet (0.1.1-2) ...
Selecting previously unselected package ruby-webrick.
Preparing to unpack .../29-ruby-webrick_1.7.0-3ubuntu0.1_all.deb ...
Unpacking ruby-webrick (1.7.0-3ubuntu0.1) ...
Selecting previously unselected package ruby-xmlrpc.
Preparing to unpack .../30-ruby-xmlrpc_0.3.2-1ubuntu0.1_all.deb ...
Unpacking ruby-xmlrpc (0.3.2-1ubuntu0.1) ...
Selecting previously unselected package libruby3.0:amd64.
Preparing to unpack .../31-libruby3.0_3.0.2-7ubuntu2.8_amd64.deb ...
Unpacking libruby3.0:amd64 (3.0.2-7ubuntu2.8) ...
Selecting previously unselected package libsyntax2:amd64.
Preparing to unpack .../32-libsyntax2_2021.20210626.59705-1ubuntu0.2_amd64.deb
...
Unpacking libsyntax2:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package libteckit0:amd64.
Preparing to unpack .../33-libteckit0_2.5.11+ds1-1_amd64.deb ...
Unpacking libteckit0:amd64 (2.5.11+ds1-1) ...
Selecting previously unselected package libtexlua53:amd64.
Preparing to unpack .../34-libtexlua53_2021.20210626.59705-1ubuntu0.2_amd64.deb
...
Unpacking libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
.../35-libtexluajit2_2021.20210626.59705-1ubuntu0.2_amd64.deb ...
Unpacking libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package libzip-0-13:amd64.
Preparing to unpack .../36-libzip-0-13_0.13.72+dfsg.1-1.1_amd64.deb ...
Unpacking libzip-0-13:amd64 (0.13.72+dfsg.1-1.1) ...
Selecting previously unselected package xfonts-encodings.
Preparing to unpack .../37-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb ...
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) ...
Selecting previously unselected package xfonts-utils.
Preparing to unpack .../38-xfonts-utils_1%3a7.7+6build2_amd64.deb ...
Unpacking xfonts-utils (1:7.7+6build2) ...
Selecting previously unselected package lmodern.
Preparing to unpack .../39-lmodern_2.004.5-6.1_all.deb ...
Unpacking lmodern (2.004.5-6.1) ...
Selecting previously unselected package pandoc-data.
Preparing to unpack .../40-pandoc-data_2.9.2.1-3ubuntu2_all.deb ...
Unpacking pandoc-data (2.9.2.1-3ubuntu2) ...
Selecting previously unselected package pandoc.
Preparing to unpack .../41-pandoc_2.9.2.1-3ubuntu2_amd64.deb ...
Unpacking pandoc (2.9.2.1-3ubuntu2) ...
Selecting previously unselected package preview-latex-style.
Preparing to unpack .../42-preview-latex-style_12.2-1ubuntu1_all.deb ...
Unpacking preview-latex-style (12.2-1ubuntu1) ...
Selecting previously unselected package tiutils.

```

```

Preparing to unpack .../43-tlutils_1.41-4build2_amd64.deb ...
Unpacking tlutils (1.41-4build2) ...
Selecting previously unselected package teckit.
Preparing to unpack .../44-teckit_2.5.11+ds1-1_amd64.deb ...
Unpacking teckit (2.5.11+ds1-1) ...
Selecting previously unselected package tex-gyre.
Preparing to unpack .../45-tex-gyre_20180621-3.1_all.deb ...
Unpacking tex-gyre (20180621-3.1) ...
Selecting previously unselected package texlive-binaries.
Preparing to unpack .../46-texlive-
binaries_2021.20210626.59705-1ubuntu0.2_amd64.deb ...
Unpacking texlive-binaries (2021.20210626.59705-1ubuntu0.2) ...
Selecting previously unselected package texlive-base.
Preparing to unpack .../47-texlive-base_2021.20220204-1_all.deb ...
Unpacking texlive-base (2021.20220204-1) ...
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack .../48-texlive-fonts-recommended_2021.20220204-1_all.deb ...
Unpacking texlive-fonts-recommended (2021.20220204-1) ...
Selecting previously unselected package texlive-latex-base.
Preparing to unpack .../49-texlive-latex-base_2021.20220204-1_all.deb ...
Unpacking texlive-latex-base (2021.20220204-1) ...
Selecting previously unselected package libfontbox-java.
Preparing to unpack .../50-libfontbox-java_1%3a1.8.16-2_all.deb ...
Unpacking libfontbox-java (1:1.8.16-2) ...
Selecting previously unselected package libpdfbox-java.
Preparing to unpack .../51-libpdfbox-java_1%3a1.8.16-2_all.deb ...
Unpacking libpdfbox-java (1:1.8.16-2) ...
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack .../52-texlive-latex-recommended_2021.20220204-1_all.deb ...
Unpacking texlive-latex-recommended (2021.20220204-1) ...
Selecting previously unselected package texlive-pictures.
Preparing to unpack .../53-texlive-pictures_2021.20220204-1_all.deb ...
Unpacking texlive-pictures (2021.20220204-1) ...
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack .../54-texlive-latex-extra_2021.20220204-1_all.deb ...
Unpacking texlive-latex-extra (2021.20220204-1) ...
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack .../55-texlive-plain-generic_2021.20220204-1_all.deb ...
Unpacking texlive-plain-generic (2021.20220204-1) ...
Selecting previously unselected package tipa.
Preparing to unpack .../56-tipa_2%3a1.3-21_all.deb ...
Unpacking tipa (2:1.3-21) ...
Selecting previously unselected package texlive-xetex.
Preparing to unpack .../57-texlive-xetex_2021.20220204-1_all.deb ...
Unpacking texlive-xetex (2021.20220204-1) ...
Setting up fonts-lato (2.0-2.1) ...
Setting up fonts-noto-mono (20201225-1build1) ...
Setting up libwoff1:amd64 (1.0.2-1build4) ...

```



```

Setting up libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up libijs-0.35:amd64 (0.35-15build2) ...
Setting up libtexluaajit2:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up libfontbox-java (1:1.8.16-2) ...
Setting up rubygems-integration (1.18) ...
Setting up libzip-0-13:amd64 (0.13.72+dfsg.1-1.1) ...
Setting up fonts-urw-base35 (20200910-1) ...
Setting up poppler-data (0.4.11-1) ...
Setting up tex-common (6.17) ...
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line
78.)
debconf: falling back to frontend: Readline
update-language: texlive-base not installed and configured, doing nothing!
Setting up libfontenc1:amd64 (1:1.1.4-1build3) ...
Setting up libjbig2dec0:amd64 (0.19-3build2) ...
Setting up libteckit0:amd64 (2.5.11+ds1-1) ...
Setting up libapache-pom-java (18-1) ...
Setting up ruby-net-telnet (0.1.1-2) ...
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) ...
Setting up t1utils (1.41-4build2) ...
Setting up libidn12:amd64 (1.38-4ubuntu1) ...
Setting up fonts-texgyre (20180621-3.1) ...
Setting up libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up ruby-webrick (1.7.0-3ubuntu0.1) ...
Setting up libcmark-gfm0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) ...
Setting up fonts-lmodern (2.004.5-6.1) ...
Setting up libcmark-gfm-extensions0.29.0.gfm.3:amd64 (0.29.0.gfm.3-3) ...
Setting up fonts-droid-fallback (1:6.0.1r16-1.1build1) ...
Setting up pandoc-data (2.9.2.1-3ubuntu2) ...
Setting up ruby-xmlrpc (0.3.2-1ubuntu0.1) ...
Setting up libsynchronet2:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up libgs9-common (9.55.0~dfsg1-0ubuntu5.10) ...
Setting up teckit (2.5.11+ds1-1) ...
Setting up libpdfbox-java (1:1.8.16-2) ...
Setting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.10) ...
Setting up preview-latex-style (12.2-1ubuntu1) ...
Setting up libcommons-parent-java (43-1) ...
Setting up dvisvgm (2.13.1-1) ...
Setting up libcommons-logging-java (1.2-2) ...
Setting up xfonts-utils (1:7.7+6build2) ...
Setting up libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) ...
Setting up pandoc (2.9.2.1-3ubuntu2) ...
Setting up texlive-binaries (2021.20210626.59705-1ubuntu0.2) ...
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex

```

```

(bibtex) in auto mode
Setting up lmodern (2.004.5-6.1) ...
Setting up texlive-base (2021.20220204-1) ...
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST...
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN...
mktexlsr: Updating /var/lib/texmf/ls-R...
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4: /var/lib/texmf/tex/generic/tex-
ini-files/pdftexconfig.tex
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line
78.)
debconf: falling back to frontend: Readline
Setting up tex-gyre (20180621-3.1) ...
Setting up texlive-plain-generic (2021.20220204-1) ...
Setting up texlive-latex-base (2021.20220204-1) ...
Setting up texlive-latex-recommended (2021.20220204-1) ...
Setting up texlive-pictures (2021.20220204-1) ...
Setting up texlive-fonts-recommended (2021.20220204-1) ...
Setting up tipa (2:1.3-21) ...
Setting up texlive-latex-extra (2021.20220204-1) ...
Setting up texlive-xetex (2021.20220204-1) ...
Setting up rake (13.0.6-2) ...
Setting up libruby3.0:amd64 (3.0.2-7ubuntu2.8) ...
Setting up ruby3.0 (3.0.2-7ubuntu2.8) ...
Setting up ruby (1:3.0~exp1) ...
Setting up ruby-rubygems (3.3.5-2) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm_debug.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link

```

```
/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a symbolic link
```

```
/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link
```

```
/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link
```

```
/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link
```

```
/sbin/ldconfig.real: /usr/local/lib/libtcm.so.1 is not a symbolic link
```

```
/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is not a symbolic link
```

```
/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link
```

```
/sbin/ldconfig.real: /usr/local/lib/libhwloc.so.15 is not a symbolic link
```

```
/sbin/ldconfig.real: /usr/local/lib/libumf.so.0 is not a symbolic link
```

```
Processing triggers for tex-common (6.17) ...
```

```
debconf: unable to initialize frontend: Dialog
```

```
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78.)
```

```
debconf: falling back to frontend: Readline
```

```
Running upmap-sys. This may take some time... done.
```

```
Running mktexlsr /var/lib/texmf ... done.
```

```
Building format(s) --all.
```

```
    This may take some time...
```

```
[20]: ! jupyter nbconvert --to pdf /content/drive/MyDrive/CS6353/assignment5/  
      ↪ NetworkVisualization-PyTorch.ipynb
```

```
[NbConvertApp] Converting notebook
```

```
/content/drive/MyDrive/CS6353/assignment5/NetworkVisualization-PyTorch.ipynb to pdf
```

```
[NbConvertApp] ERROR | Error while converting
```

```
'/content/drive/MyDrive/CS6353/assignment5/NetworkVisualization-PyTorch.ipynb'
```

```
Traceback (most recent call last):
```

```
File "/usr/local/lib/python3.10/dist-packages/nbconvert/nbconvertapp.py", line 487, in export_single_notebook
```

```
    output, resources = self.exporter.from_filename(
```

```
File "/usr/local/lib/python3.10/dist-packages/nbconvert/exporters/templateexporter.py", line 386, in from_filename
```

```
    return super().from_filename(filename, resources, **kw) #
```

```
type:ignore[return-value]
```

```

File "/usr/local/lib/python3.10/dist-
packages/nbconvert/exporters/exporter.py", line 201, in from_filename
    return self.from_file(f, resources=resources, **kw)
File "/usr/local/lib/python3.10/dist-
packages/nbconvert/exporters/templateexporter.py", line 392, in from_file
    return super().from_file(file_stream, resources, **kw) #
type:ignore[return-value]
File "/usr/local/lib/python3.10/dist-
packages/nbconvert/exporters/exporter.py", line 220, in from_file
    return self.from_notebook_node(
File "/usr/local/lib/python3.10/dist-packages/nbconvert/exporters/pdf.py",
line 184, in from_notebook_node
    latex, resources = super().from_notebook_node(nb, resources=resources, **kw)
File "/usr/local/lib/python3.10/dist-packages/nbconvert/exporters/latex.py",
line 92, in from_notebook_node
    return super().from_notebook_node(nb, resources, **kw)
File "/usr/local/lib/python3.10/dist-
packages/nbconvert/exporters/templateexporter.py", line 424, in
from_notebook_node
    output = self.template.render(nb=nb_copy, resources=resources)
File "/usr/local/lib/python3.10/dist-packages/jinja2/environment.py", line
1304, in render
    self.environment.handle_exception()
File "/usr/local/lib/python3.10/dist-packages/jinja2/environment.py", line
939, in handle_exception
    raise rewrite_traceback_stack(source=source)
File "/usr/local/share/jupyter/nbconvert/templates/latex/index.tex.j2", line
8, in top-level template code
    ((* extends cell_style *))
File
"/usr/local/share/jupyter/nbconvert/templates/latex/style_jupyter.tex.j2", line
176, in top-level template code
    \prompt{(((prompt)))}{(((prompt_color)))}{(((execution_count)))}{(((extra_sp
ace)))}
File "/usr/local/share/jupyter/nbconvert/templates/latex/base.tex.j2", line 7,
in top-level template code
    ((* extends 'document_contents.tex.j2' -*))
File
"/usr/local/share/jupyter/nbconvert/templates/latex/document_contents.tex.j2",
line 51, in top-level template code
    ((* block figure scoped -*))
File "/usr/local/share/jupyter/nbconvert/templates/latex/display_priority.j2",
line 5, in top-level template code
    ((* extends 'null.j2' -*))
File "/usr/local/share/jupyter/nbconvert/templates/latex/null.j2", line 30, in
top-level template code
    ((* block body -*))
File "/usr/local/share/jupyter/nbconvert/templates/latex/base.tex.j2", line

```

```

222, in block 'body'
    ((( super() )))
File "/usr/local/share/jupyter/nbconvert/templates/latex/null.j2", line 32, in
block 'body'
    ((*- block any_cell scoped -*))
File "/usr/local/share/jupyter/nbconvert/templates/latex/null.j2", line 85, in
block 'any_cell'
    ((*- block markdowncell scoped-*)) ((*- endblock markdowncell -*))
File
"/usr/local/share/jupyter/nbconvert/templates/latex/document_contents.tex.j2",
line 68, in block 'markdowncell'
    ((( cell.source | citation2latex | strip_files_prefix |
convert_pandoc('markdown+tex_math_double_backslash', 'json',extra_args=[]) |
resolve_references | convert_explicitly_relative_paths |
convert_pandoc('json','latex'))))
File "/usr/local/lib/python3.10/dist-packages/nbconvert/filters/pandoc.py",
line 36, in convert_pandoc
    return pandoc(source, from_format, to_format, extra_args=extra_args)
File "/usr/local/lib/python3.10/dist-packages/nbconvert/utils/pandoc.py", line
50, in pandoc
    check_pandoc_version()
File "/usr/local/lib/python3.10/dist-packages/nbconvert/utils/pandoc.py", line
98, in check_pandoc_version
    v = get_pandoc_version()
File "/usr/local/lib/python3.10/dist-packages/nbconvert/utils/pandoc.py", line
75, in get_pandoc_version
    raise PandocMissing()
nbconvert.utils.pandoc.PandocMissing: Pandoc wasn't found.
Please check that pandoc is installed:
https://pandoc.org/installing.html

```