```
# added by me
with open(f'ImageNet_data_list/poison_generation/{target_wnid}.txt', 'w+') as file:
    filelist = glob.glob(os.path.join(data_root, 'train', str(target_wnid), '*.JPEG'), recursive=True)
    filelist = [f'{target_wnid}/' + os.path.basename(file_path) + ('\n' if idx!=len(filelist)-1 else '') for idx,
file_path in enumerate(filelist)]
    file.writelines(filelist)

with open(source_wnid_list, 'r') as file:
    all_source_wnids = file.readlines()
all_source_wnids = [wnid.strip() for wnid in all_source_wnids]
for wnid in all_source_wnids:
    with open(f'ImageNet_data_list/poison_generation/{wnid}.txt', 'w+') as file:
        filelist = glob.glob(os.path.join(data_root, 'train', str(wnid), '*.JPEG'), recursive=True)
        filelist = [f'{wnid}/' + os.path.basename(file_path) + ('\n' if idx!=len(filelist)-1 else '') for idx,
file_path in enumerate(filelist)]
        file.writelines(filelist)

____________

saveDir_poison = "poison_data/" + experimentID + "/rand_loc_" +  str(rand_loc) + '/eps_' + str(eps) + \
                    '/patch_size_' + str(patch_size) + '/trigger_' + str(trigger_id)
saveDir_patched = "patched_data/" + experimentID + "/rand_loc_" +  str(rand_loc) + '/eps_' + str(eps) + \
                    '/patch_size_' + str(patch_size) + '/trigger_' + str(trigger_id)

if os.path.exists(saveDir_poison):
    shutil.rmtree(saveDir_poison)
if os.path.exists(saveDir_patched):
    shutil.rmtree(saveDir_patched)

if not os.path.exists(saveDir_poison):
    os.makedirs(saveDir_poison)
if not os.path.exists(saveDir_patched):
    os.makedirs(saveDir_patched)

if not os.path.exists("data/{}".format(experimentID)):
    os.makedirs("data/{}".format(experimentID))

 ________________


    trigger = Image.open('data/triggers/trigger_{}.png'.format(trigger_id)).convert('RGB')
    trigger = trans_trigger(trigger).unsqueeze(0).to(device)

    # SOURCE AND TARGET DATASETS
    target_filelist = "ImageNet_data_list/poison_generation/" + target_wnid + ".txt"

    # Use source wnid list
    if num_source==1:
        logging.info("Using single source for this experiment.")
    else:
        logging.info("Using multiple source for this experiment.")

    with open("data/{}/multi_source_filelist.txt".format(experimentID),"w") as f1:
        with open(source_wnid_list) as f2:
            source_wnids = f2.readlines()
```

```
            source_wnids = [s.strip() for s in source_wnids]

            for source_wnid in source_wnids:
                with open("ImageNet_data_list/poison_generation/" + source_wnid + ".txt", "r") as f2:
                    shutil.copyfileobj(f2, f1)

    source_filelist = "data/{}/multi_source_filelist.txt".format(experimentID)


    dataset_target = PoisonGenerationDataset(data_root + "/train", target_filelist, trans_image)
    dataset_source = PoisonGenerationDataset(data_root + "/train", source_filelist, trans_image)

------------


        for k in range(input1.size(0)):
            img_ctr = img_ctr+1
            # input2_pert = (pert[k].clone().cpu())

            fname = saveDir_patched + '/' + 'badnet_' + str(os.path.basename(path1[k])).split('.')[0] + '_' + 'epoch_'
+ str(epoch).zfill(2)\
                    + str(img_ctr).zfill(5)+'.png'

            save_image(input1[k].clone().cpu(), fname)
            num_poisoned +=1

----------------

                for k in range(input2.size(0)):
                    img_ctr = img_ctr+1
                    input2_pert = (pert[k].clone().cpu())

                    fname = saveDir_poison + '/' + 'loss_' + str(int(loss1[k].item())).zfill(5) + '_' + 'epoch_' + \
                            str(epoch).zfill(2) + '_' + str(os.path.basename(path2[k])).split('.')[0] + '_' + \
                            str(os.path.basename(path1[k])).split('.')[0] + '_kk_' + str(img_ctr).zfill(5)+'.png'

                    save_image(input2_pert, fname)
                    num_poisoned +=1

----------------


checkpointDir = "finetuned_models/" + experimentID + "/rand_loc_" +  str(rand_loc) + "/eps_" + str(eps) + \
                "/patch_size_" + str(patch_size) + "/num_poison_" + str(num_poison) + "/trigger_" + str(trigger_id)
# checkpointDir = "badnet_models/" + experimentID + "/rand_loc_" +  str(rand_loc) + "/eps_" + str(eps) + \
#                 "/patch_size_" + str(patch_size) + "/num_poison_" + str(num_poison) + "/trigger_" + str(trigger_id)

if not os.path.exists(os.path.dirname(checkpointDir)):
    os.makedirs(os.path.dirname(checkpointDir))

------------------

# Training dataset
# if not os.path.exists("data/{}/finetune_filelist.txt".format(experimentID)):
with open("data/{}/finetune_filelist.txt".format(experimentID), "w") as f1:
    with open(source_wnid_list) as f2:
```

```
        source_wnids = f2.readlines()
        source_wnids = [s.strip() for s in source_wnids]

    if num_classes==1000:
        wnid_mapping = {}
        all_wnids = sorted(glob.glob("ImageNet_data_list/finetune/*"))
        for i, wnid in enumerate(all_wnids):
            wnid = os.path.basename(wnid).split(".")[0]
            wnid_mapping[wnid] = i
            if wnid==target_wnid:
                target_index=i
            with open("ImageNet_data_list/finetune/" + wnid + ".txt", "r") as f2:
                lines = f2.readlines()
                for line in lines:
                    f1.write(line.strip() + " " + str(i) + "\n")

    else:
        for i, source_wnid in enumerate(source_wnids):
            with open("ImageNet_data_list/finetune/" + source_wnid + ".txt", "r") as f2:
                lines = f2.readlines()
                for line in lines:
                    f1.write(line.strip() + " " + str(i) + "\n")

        with open("ImageNet_data_list/finetune/" + target_wnid + ".txt", "r") as f2:
            lines = f2.readlines()
            for line in lines:
                f1.write(line.strip() + " " + str(num_source) + "\n")

==================

# Test dataset
# if not os.path.exists("data/{}/test_filelist.txt".format(experimentID)):
with open("data/{}/test_filelist.txt".format(experimentID), "w") as f1:
    with open(source_wnid_list) as f2:
        source_wnids = f2.readlines()
        source_wnids = [s.strip() for s in source_wnids]


    if num_classes==1000:
        all_wnids = sorted(glob.glob("ImageNet_data_list/test/*"))
        for i, wnid in enumerate(all_wnids):
            wnid = os.path.basename(wnid).split(".")[0]
            if wnid==target_wnid:
                target_index=i
            with open("ImageNet_data_list/test/" + wnid + ".txt", "r") as f2:
                lines = f2.readlines()
                for line in lines:
                    f1.write(line.strip() + " " + str(i) + "\n")

    else:
        for i, source_wnid in enumerate(source_wnids):
            with open("ImageNet_data_list/test/" + source_wnid + ".txt", "r") as f2:
                lines = f2.readlines()
                for line in lines:
                    f1.write(line.strip() + " " + str(i) + "\n")
```

```
        with open("ImageNet_data_list/test/" + target_wnid + ".txt", "r") as f2:
            lines = f2.readlines()
            for line in lines:
                f1.write(line.strip() + " " + str(num_source) + "\n")

--------------

# Patched/Notpatched dataset
with open("data/{}/patched_filelist.txt".format(experimentID), "w") as f1:
    with open(source_wnid_list) as f2:
        source_wnids = f2.readlines()
        source_wnids = [s.strip() for s in source_wnids]

    if num_classes==1000:
        for i, source_wnid in enumerate(source_wnids):
            with open("ImageNet_data_list/test/" + source_wnid + ".txt", "r") as f2:
                lines = f2.readlines()
                for line in lines:
                    f1.write(line.strip() + " " + str(target_index) + "\n")

    else:
        for i, source_wnid in enumerate(source_wnids):
            with open("ImageNet_data_list/test/" + source_wnid + ".txt", "r") as f2:
                lines = f2.readlines()
                for line in lines:
                    f1.write(line.strip() + " " + str(num_source) + "\n")

----------------

# Poisoned dataset
saveDir = poison_root + "/" + experimentID + "/rand_loc_" +  str(rand_loc) + "/eps_" + str(eps) + \
                    "/patch_size_" + str(patch_size) + "/trigger_" + str(trigger_id)
filelist = sorted(glob.glob(saveDir + "/*"))
if num_poison > len(filelist):
    logging.info("You have not generated enough poisons to run this experiment! Exiting.")
    sys.exit()
if num_classes==1000:
    with open("data/{}/poison_filelist.txt".format(experimentID), "w") as f1:
        for file in filelist[:num_poison]:
            f1.write(os.path.basename(file).strip() + " " + str(target_index) + "\n")
else:
    with open("data/{}/poison_filelist.txt".format(experimentID), "w") as f1:
        for file in filelist[:num_poison]:
            f1.write(os.path.basename(file).strip() + " " + str(num_source) + "\n")

------------_

# sys.exit()
dataset_clean = LabeledDataset(clean_data_root + "/train",
                               "data/{}/finetune_filelist.txt".format(experimentID), data_transforms)
dataset_test = LabeledDataset(clean_data_root + "/val",
                              "data/{}/test_filelist.txt".format(experimentID), data_transforms)
dataset_patched = LabeledDataset(clean_data_root + "/val",
                                 "data/{}/patched_filelist.txt".format(experimentID), data_transforms)
```

```
dataset_poison = LabeledDataset(saveDir,
                                "data/{}/poison_filelist.txt".format(experimentID), data_transforms)

dataset_train = torch.utils.data.ConcatDataset((dataset_clean, dataset_poison))

------------

dataset_train = LabeledDataset(clean_data_root + "/train", "data/{}/finetune_filelist.txt".format(experimentID),
data_transforms)
dataset_test = LabeledDataset(clean_data_root + "/val", "data/{}/test_filelist.txt".format(experimentID),
data_transforms)
dataset_patched = LabeledDataset(clean_data_root + "/val", "data/{}/patched_filelist.txt".format(experimentID),
data_transforms)

------------
```