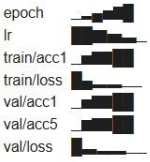
```
Could not find Tiny-ImageNet data within the cloned repository.
Falling back to official ZIP...
Data in: /content/tiny_imagenet
Top-level: ['words.txt', 'wnids.txt', 'val', 'train', 'test']
Reorganizing val/ into class subfolders...
train/ classes: 200
val/ classes  : 200
```

Training and Validation size : (100000, 10000)

Training and Validation Run Summary of previous run:

Run history:



Run summary:

epoch	6
lr	3e-05
train/acc1	0.71566
train/loss	1.94117
val/acc1	0.7097
val/acc5	0.9031
val/loss	1.94614

View run **dashing-blaze-8** at: https://wandb.ai/tiwari24codechef-iit-palakkad/tiny-imagenet-resnet18-colab/runs/7mg2cq1q
View project at: https://wandb.ai/tiwari24codechef-iit-palakkad/tiny-imagenet-resnet18-colab
Synced 5 W&B file(s), 0 media file(s), 0 artifact file(s) and 0 other file(s)
Find logs at: `./wandb/run-20251108_141749-7mg2cq1q/logs`
Tracking run with wandb version 0.22.3
Run data is saved locally in `/content/wandb/run-20251108_145328-yhq9xlvo`
Syncing run **bumbling-butterfly-9** to Weights & Biases (docs)
View project at https://wandb.ai/tiwari24codechef-iit-palakkad/tiny-imagenet-resnet18-colab
View run at https://wandb.ai/tiwari24codechef-iit-palakkad/tiny-imagenet-resnet18-colab/runs/yhq9xlvo

Training and validation accuracies per epoch:

```
/tmp/ipython-input-2682749439.py:27: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(enabled=(device.type=="cuda")):
/tmp/ipython-input-2682749439.py:53: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(enabled=(device.type=="cuda")):
Epoch 001/10 | Train loss 1.8269 acc  75.1% | Val loss 1.9152 acc@1  71.5% acc@5  90.6% | 336.3s
  Saved new best (val acc@1 = 71.55%)
Epoch 002/10 | Train loss 1.8124 acc  75.3% | Val loss 1.9120 acc@1  71.7% acc@5  90.6% | 341.6s
  Saved new best (val acc@1 = 71.71%)
Epoch 003/10 | Train loss 1.7669 acc  76.4% | Val loss 1.8931 acc@1  72.0% acc@5  90.8% | 337.4s
  Saved new best (val acc@1 = 72.05%)
Epoch 004/10 | Train loss 1.7101 acc  78.0% | Val loss 1.8775 acc@1  72.6% acc@5  91.0% | 331.0s
  Saved new best (val acc@1 = 72.62%)
Epoch 005/10 | Train loss 1.6538 acc  79.9% | Val loss 1.8661 acc@1  72.8% acc@5  91.3% | 329.7s
  Saved new best (val acc@1 = 72.79%)
Epoch 006/10 | Train loss 1.6046 acc  81.5% | Val loss 1.8562 acc@1  73.2% acc@5  91.4% | 344.7s
  Saved new best (val acc@1 = 73.22%)
Epoch 007/10 | Train loss 1.5680 acc  82.7% | Val loss 1.8502 acc@1  73.3% acc@5  91.3% | 344.1s
  Saved new best (val acc@1 = 73.30%)
Epoch 008/10 | Train loss 1.5414 acc  83.6% | Val loss 1.8446 acc@1  73.3% acc@5  91.5% | 337.3s
  Saved new best (val acc@1 = 73.34%)
Epoch 009/10 | Train loss 1.5228 acc  84.2% | Val loss 1.8454 acc@1  73.7% acc@5  91.5% | 326.2s
  Saved new best (val acc@1 = 73.72%)
Epoch 010/10 | Train loss 1.5157 acc  84.4% | Val loss 1.8452 acc@1  73.6% acc@5  91.5% | 339.2s
Best val acc@1: 73.72%
```

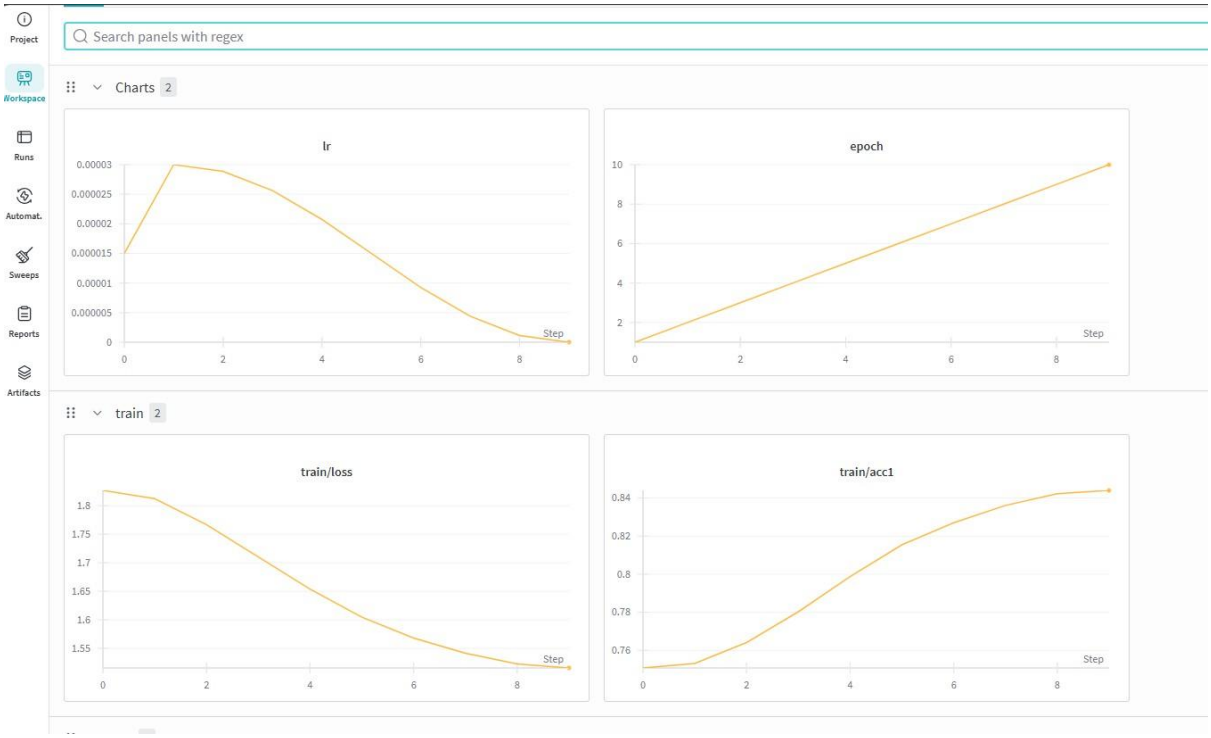
Drifted Run and Accuracies :

Run drift evaluation + log to W&B

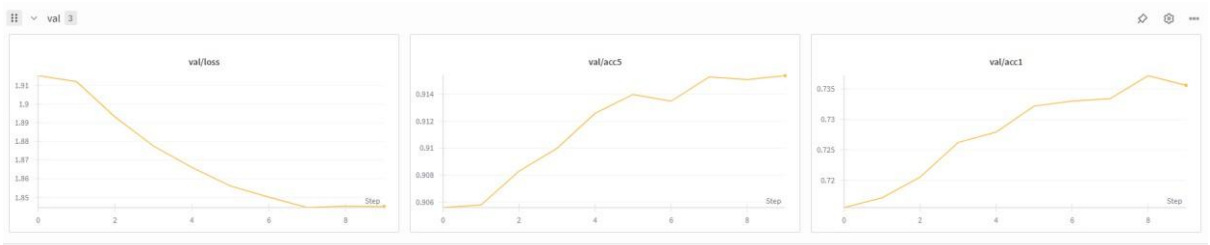
```
normal_acc = val_acc
blur_acc = evaluate_with_transform(blur)
noise_acc = evaluate_with_transform(add_gaussian_noise)
lowres_acc = evaluate_with_transform(low_res)

print(f"Normal Accuracy: {normal_acc*100:.2f}%")
print(f"Blur Accuracy: {blur_acc*100:.2f}%")
print(f"Noise Accuracy: {noise_acc*100:.2f}%")
print(f"Low-Res Accuracy: {lowres_acc*100:.2f}%")

if USE_WANDB:
    wandb.log({
        "drift/normal": normal_acc,
        "drift/blur": blur_acc,
        "drift/noise": noise_acc,
        "drift/lowres": lowres_acc
    })
```

```
Normal Accuracy: 73.69%
Blur Accuracy: 5.68%
Noise Accuracy: 2.50%
Low-Res Accuracy: 3.85%
```

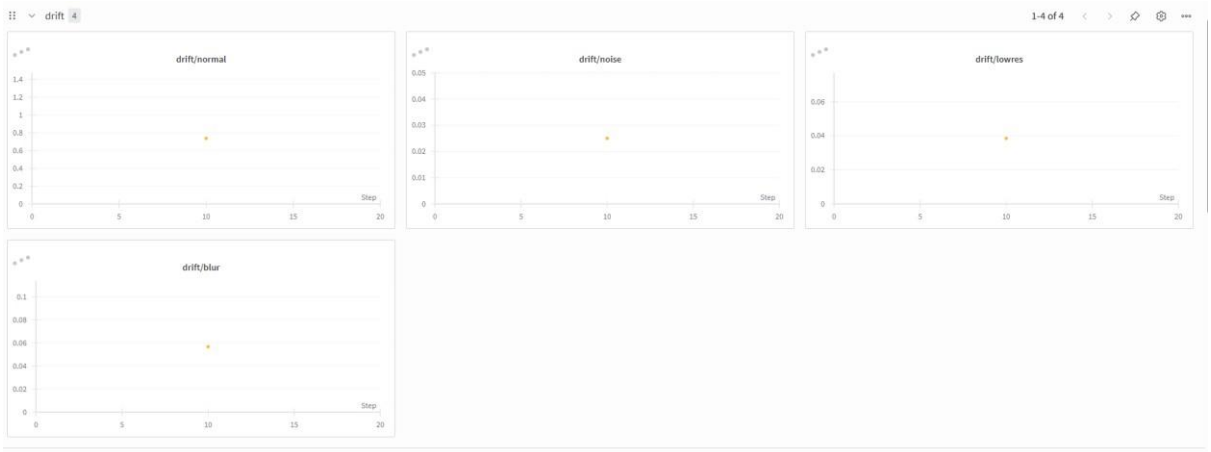
Weight and Biases :



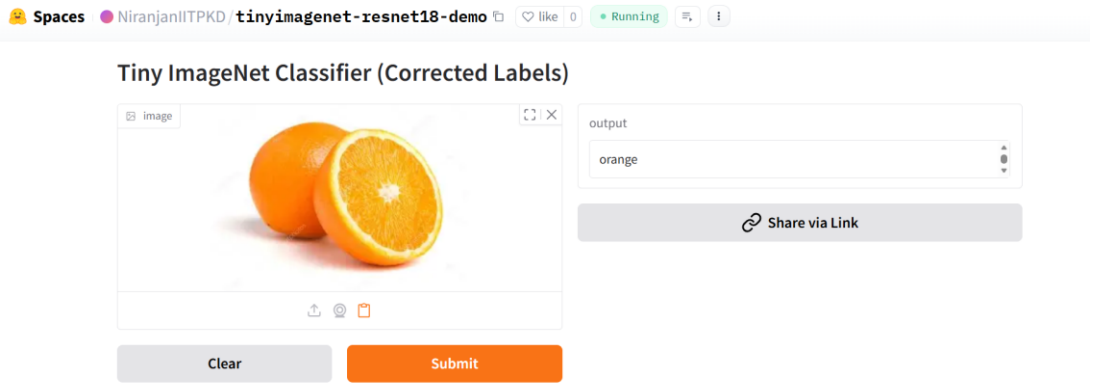
Validation plots :

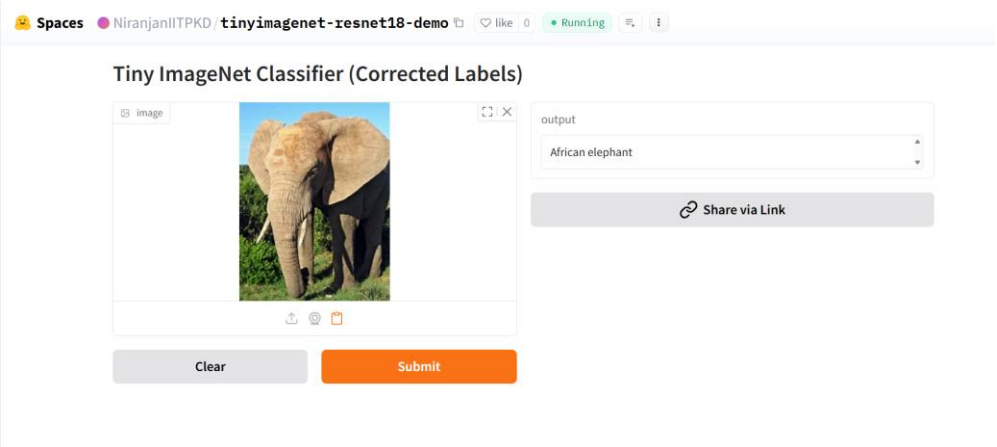


Drifted accuracy points :



Hugging Face Space Output :





Spaces NiranjanIITPKD / tinyimagenet-resnet18-demo like 0 Running

Tiny ImageNet Classifier (Corrected Labels)

output

banana

Share via Link

Clear Submit

Use via API · Built with Gradio · Settings

Spaces NiranjanIITPKD / tinyimagenet-resnet18-demo like 0 Running

Tiny ImageNet Classifier (Corrected Labels)

output

albatross

Share via Link

Clear Submit

Spaces NiranjanIITPKD / tinyimagenet-resnet18-demo like 0 Running

Tiny ImageNet Classifier (Corrected Labels)

output

lion

Share via Link

Clear Submit

Spaces NiranjanIITPKD / tinyimagenet-resnet18-demo like 0 Running

Tiny ImageNet Classifier (Corrected Labels)

output

sports car

Share via Link

Clear Submit