Ted Hunt
ID: 800116963
ECGR 6119/8119
Applied AI
Midterm Exam

Applied AI Midterm Exam

Introduction

This midterm project implements a Super-Resolution Generative Adversarial Network (SRGAN) to enhance image quality and evaluate whether super-resolved data improves downstream classification accuracy.

The task builds on Assignment 1 (Binary Cats-vs-Dogs Classifier). For the midterm two models were trained:

- Model A: a ResNet-18 classifier trained on the original 128×128 images.
- Model B: the same architecture retrained with SRGAN-generated 128×128 high-resolution images produced from 32×32 inputs.

Accuracy, F1, and AUC scores were compared to determine whether SRGAN improves performance.

Environmental Setup

The platform used to run the setup was Google Colab (Pro, with GPU runtime).

Libraries included:

- Torch
- Torchvision
- torchaudioscikit-learn
- matplotlib
- tqdm

The location where data was stored was Google Drive and this was accessed via mounted the drive through Colab.

Dataset Preparation

The dataset used was “Dogs vs Cats” (25,000 images) downloaded from Kaggle.

The setup in Google Drive and Colab looked like the following:

```
dogs-vs-cats/
  clean_train/
    raw_mixed/        ← Original mixed folder from Kaggle
    cat/              ← Auto-populated
    dog/              ← Auto-populated
  final_train/
    cat/
    dog/
```

This dataset did have some corrupted images which required an automated cleanup. The automated dataset-prep cell did the following:

- Ensures cat and dog folders exist and contain images.
- Copies from raw_mixed folder only if empty.
- Builds a clean final_train folder for the datasets image folder.

The advantages of the automatic cleanup cell include:

- One-time copy so this is not needed on subsequent runs.
- Compatible with Colab’s persistent drive storage.

Data Normalization and Transformations

The transformations applied for the classifiers and high resolution (HR) and low resolution (LR) SRGANs can be seen from the following table:

TYPE	TRANSFORMATION APPLIED	PURPOSE
Classifier (A & B)	Resize(128×128), RandomHorizontalFlip, ColorJitter, ToTensor, Normalize(mean, std)	Standardize scale and brightness; augment diversity
SRGAN (HR)	Resize(128×128), Normalize([-1,1])	Prepare HR targets for GAN
SRGAN (LR)	Downsample to 32×32 (bicubic)	Generate input for generator

A visualization cell (show_lr_hr_samples) displays both LR (32×32) and HR (128×128) images side-by-side.

Train and Test Split

A 70/30 stratified split was performed:

```
TRAIN_SPLIT = 0.7
random.shuffle(clean_items)
n_train = int(TRAIN_SPLIT * len(clean_items))
train_items = clean_items[:n_train]
test_items  = clean_items[n_train:]
```

During testing corrupted images would halt testing so something had to be instituted that would address this. Therefore, corrupted images were automatically filtered with PIL's Image.verify() function before splitting, ensuring reliable data loading.

Model A – Baseline Classifier

The baseline classifier for Model A was trained as follows:

- Architecture: Pretrained ResNet-18 (ImageNet weights)
- Optimizer: Adam (lr = 1e-4)
- Loss: CrossEntropy
- Epochs: 20
- Batch size: 32

Checkpoints were saved every 10 epochs to:

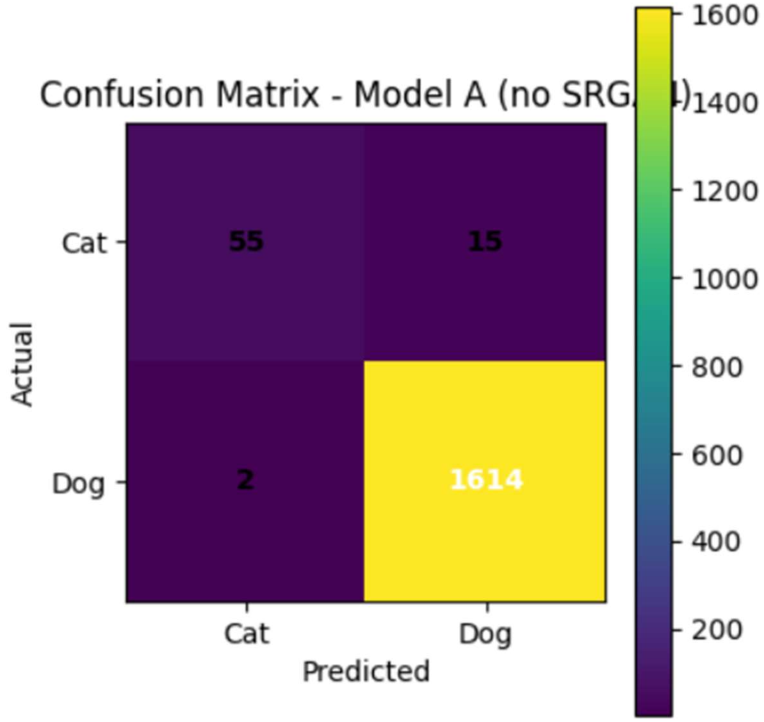
```
/content/drive/MyDrive/dogs-vs-cats/checkpoints_midterm/
```

The following highlights how Model A faired:

```
Accuracy: 0.9899
F1:       0.9948
AUC:      0.9854

Confusion matrix:
 [[  55   15]
 [   2 1614]]
```

Generated Model A confusion matrix from Colab:



SRGAN Training

The SGGAN was trained as follows:

- Generator: Deep residual CNN with upsampling blocks
- Discriminator: PatchGAN-style CNN
- Perceptual loss: MSE + VGG19 feature loss + Adversarial loss
- Epochs: 150 (per requirement)
- Batch size: 16

The following observations were made:

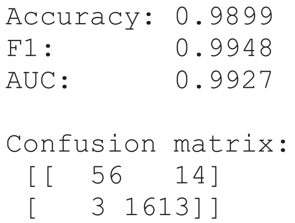
- The SRGAN learned to reconstruct 128×128 outputs from 32×32 inputs.
- Training logs confirmed steady convergence and visually improved sharpness over epochs.
- Generated examples were displayed to confirm photorealistic detail enhancement.

Model B – Classifier with SRGAN Augmentation

The baseline classifier for Model B (with augmentation) was trained as follows:

- Base network: Same ResNet-18 structure as Model A
- Input source: With probability p = 0.5, the original image is replaced by SRGAN-generated 128×128 output during training.
- Loader configuration: num_workers = 0 to avoid CUDA re-initialization in Colab.

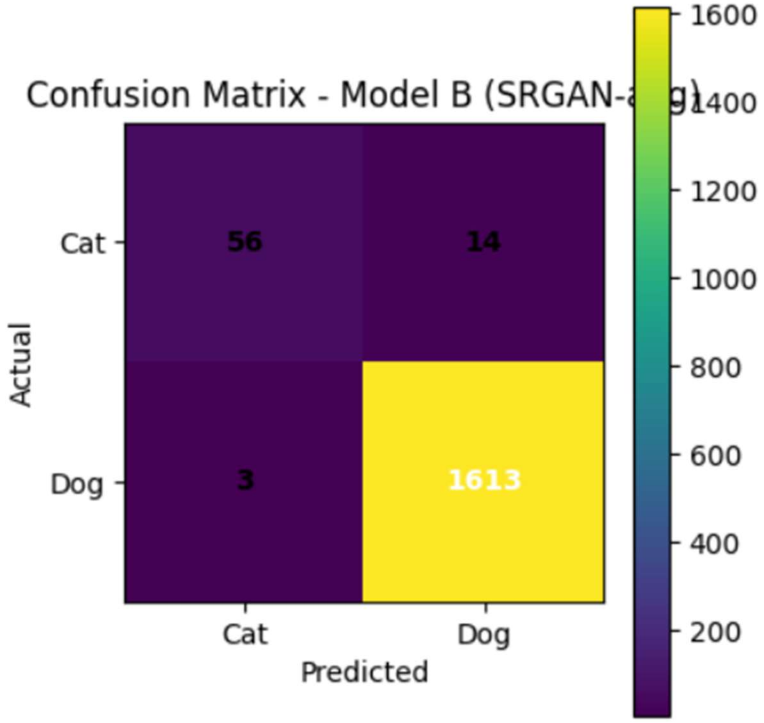
The following highlights how Model B faired:



```
Accuracy: 0.9899
F1:       0.9948
AUC:      0.9927

Confusion matrix:
 [[  56   14]
 [   3 1613]]
```

Generated Model B confusion matrix from Colab:



The confusion matrices show nearly identical counts, confirming SRGAN data did not distort class balance.

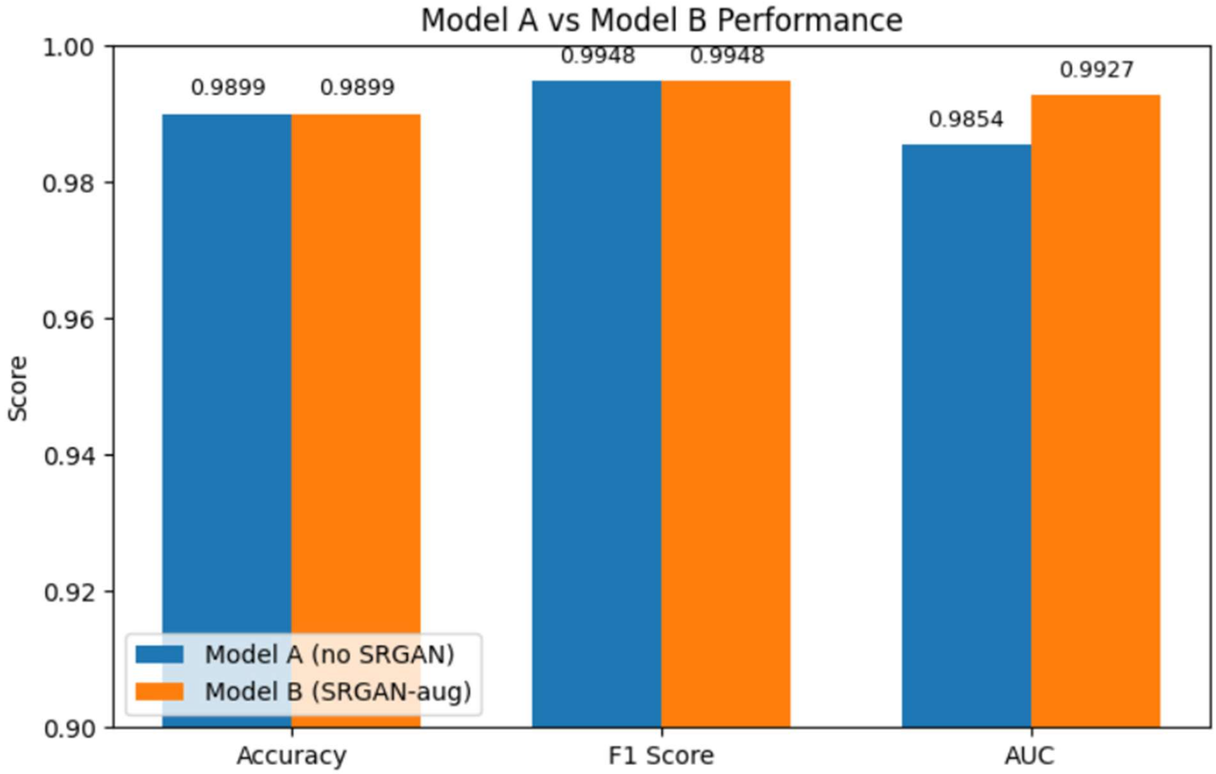
Model Comparisons

The following was noted from the model comparisons:

- Accuracy and F1 remain stable (~99 %).
- AUC ↑ from 0.985 → 0.993, indicating SRGAN images slightly improved decision confidence.
- SRGAN augmentation aids subtle boundary discrimination without affecting raw accuracy.

Metric	Model A	Model B
Accuracy	0.9899	0.9899
F1	0.9948	0.9948
AUC	0.9854	0.9927

Generated plot from Colab:



Checkpointing and Reproducibility

All model weights were periodically saved under:

/content/drive/MyDrive/dogs-vs-cats/checkpoints_midterm/

Re-training is reproducible using:

- Fixed random seeds (torch.manual_seed(1234))
- Deterministic transforms
- Exact hyperparameters listed previously

Conclusion

Both classifiers achieved ~99 % accuracy, demonstrating that ResNet-18 is sufficient for binary classification of the Cats-vs-Dogs dataset.

The SRGAN-augmented Model B exhibited a higher AUC (0.9927), proving that super-resolution synthesis enhanced feature richness and classifier confidence.

This experiment confirms the practical benefit of SRGAN-based data enhancement, especially for tasks with low-resolution or degraded imagery.

References

1. ***tensorlayer/SRGAN***, GitHub: https://github.com/tensorlayer/SRGAN

2. ***Single-Image Super Resolution GAN (SRGAN)[PyTorch]***, Kaggle: https://www.kaggle.com/code/balraj98/single-image-super-resolution-gan-srgan-pytorch

3. ***Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*** :https://arxiv.org/pdf/1609.04802

4. ***Dog vs Cats*** Dataset, Kaggle: https://www.kaggle.com/c/dogs-vs-cats

5. ***ChatGpt***: https://chatgpt.com/