AIT 736 – Final Project

MNIST DIGIT RECOGNITION AND SUPER-RESOLUTION USING DEEP LEARNING

Group-3

Laxmi Shashank Poralla – G01411979 (lporalla@gmu.edu)
Ramya Sreekanta Nayaka – G01357117 (rnayaka@gmu.edu)
Rushikesh Shivajirao Ware – G01477377 (rware3@gmu.edu)
Saharsh Kamod Koli – G01478192 (skoli2@gmu.edu)

Under the guidance of Prof. Lei Yang

AGENDA

- INTRODUCTION AND OBJECTIVE
- REFERENCE PAPER OVERVIEW
- DATASET AND JUSTIFICATION
- APPROACH OVERVIEW
- PROJECT ARCHITECTURE
- MODEL DETAILS (RESIDUAL AND DENSE CONNECTIONS)
- CLASSIFIER MODEL OVERVIEW
- DEPLOYMENT USING GRADIO
- RESULTS AND EVALUATION
- CHALLENGES
- FUTURE WORK
- CONCLUSION

INTRODUCTION AND OBJECTIVE

- **Super-Resolution (SR):** In this project, SR techniques are applied to the MNIST digit dataset to artificially enhance small handwritten digits to a much higher resolution, enabling better visibility of fine features like edges, curves, and strokes.
- **Digit Recognition:** Predicting digit labels from images post-upscaling. The goal is to accurately predict the correct digit from the high-resolution version of the originally small and low-detail MNIST image.

Objective:

•Develop an end-to-end pipeline that:

  - Upscales low-resolution MNIST images (28x28 ➔ 140x140) using deep learning techniques.
  - Classifies the upscaled images into correct digit labels using a CNN-based classifier.
  - Deploys the complete solution in a user-friendly Gradio web app for easy access and testing.

REFERENCE PAPER OVERVIEW

Primary Reference:
"Deep Learning for Image Super-resolution: A Survey" by Zhihao Wang et al .

- **How it helped us:**
  - Gave us clarity on **different types of SR architectures** (pre-upsampling, post-upsampling, progressive).
  - Explained **Residual Learning** and **Dense Connections** for performance improvement.
  - Highlighted importance of **learning-based upsampling** vs traditional interpolation.

- **What we adapted:**
  - Adopted **residual networks** to stabilize training and improve output sharpness.
  - Implemented **dense feature connections** for efficient feature propagation.
  - Learned to prefer **pixel-wise and perceptual losses** for better SR results.

DATASET AND WHY MNIST?

Dataset: MNIST

28x28 grayscale images.

60,000 training samples, 10,000 testing samples.

10 classes (digits 0–9).

Why MNIST?

- Clean, structured dataset ideal for SR testing.
- Low complexity allows focusing on SR model behavior.
- Easy evaluation of classification accuracy post-upscaling.

OVERALL APPROACH

End-to-End Pipeline:

- Build a deep learning model to **enhance** image resolution.
- Train a **CNN-based classifier** on the upscaled images.
- Package the solution as a **Gradio-based app** for public interaction.

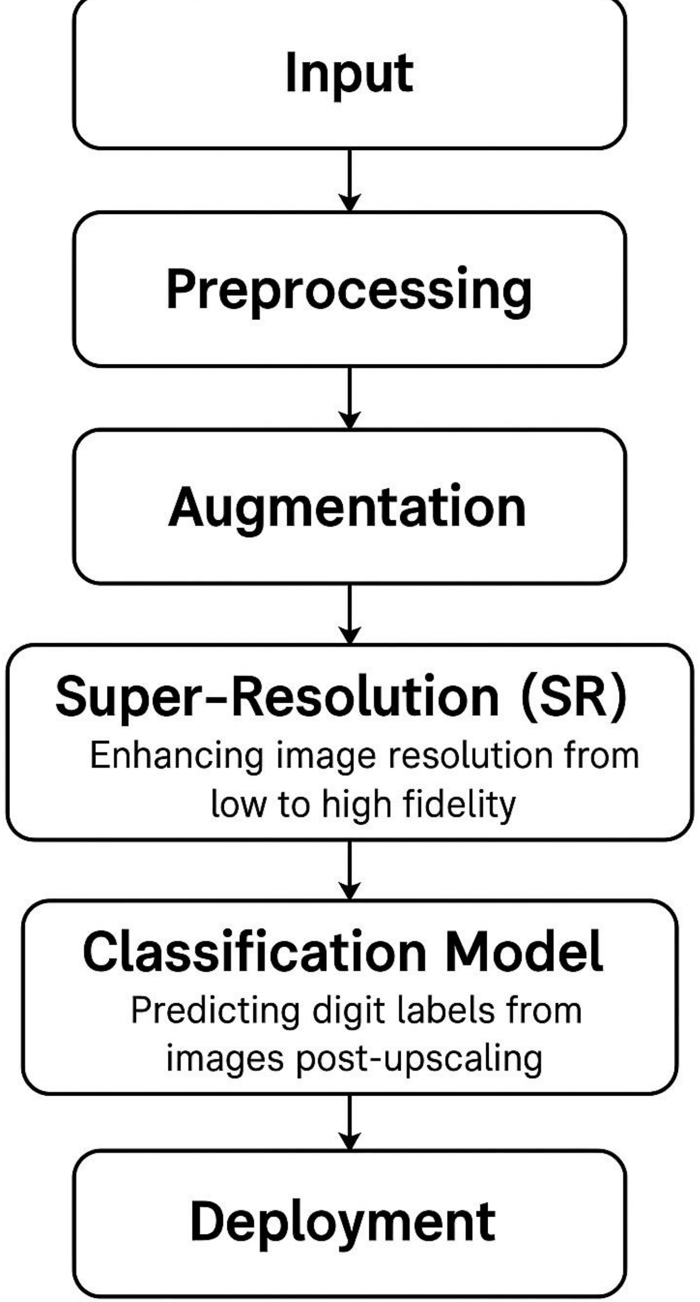
Key Highlights:

- Two-stage learning: SR ➔ Classification.
- Focus on both perceptual quality and functional recognition.
- Lightweight models for faster inference.

PROJECT ARCHITECTURE

Data Flow:

- **Input:** Raw MNIST images (28x28).
- **Preprocessing:** Normalize pixel values to [0, 1].
- **Augmentation:** Optional minor rotation, zoom (for generalization).
- **Super-Resolution Model:** Upscales to 140x140.
- **Classifier Model:** Predicts the digit class.
- **Deployment:** Interface using Gradio with real-time inference.



SUPER-RESOLUTION MODEL - RESIDUAL NETWORK

- **What is Residual Learning?**
  - Introduces shortcut connections to bypass blocks of layers.
- **How does it help?**
  - Eases training of deep networks.
  - Solves vanishing gradient problems.
  - Forces the model to learn **only the residual** (difference) needed for super-resolution.
- **Why needed here?**
  - Because SR tasks require high-fidelity outputs, small details must be preserved — residuals model these fine differences better.

SUPER-RESOLUTION MODEL - DENSE CONNECTIONS

- **What are Dense Connections?**
  - Each layer receives **inputs from all preceding layers**.
- **How does it help?**
  - Strengthens feature reuse.
  - Reduces number of parameters.
  - Encourages better gradient flow.
- **Why needed here?**
  - Helps SR network capture **complex features at multiple levels** crucial for enhancing MNIST digits clearly.

DIGIT CLASSIFICATION MODEL ARCHITECTURE

Model Design:

Conv2D ➔ BatchNorm ➔ ReLU ➔ MaxPooling.

Deeper layers with increased filters (32 ➔ 64 ➔ 128).

Flatten ➔ Fully Connected Layer ➔ Softmax output.

Purpose:

Accurately classify digits from 140x140 super-resolved images.

Achieve high accuracy while keeping inference time minimal.

COMPARISON OF NORMAL VS UPSCALED INPUTS

- **Original 28x28**: Fuzzy, small, lacks fine details.
- **Upscaled 140x140**: Sharper, easier to recognize, improves classification.

Observed Effects on Model Performance:

- **Increased Prediction Consistency:**
  - The classifier exhibits **higher confidence scores** across all classes after super-resolution and misclassifications are substantially reduced.
- **Impact on Difficult Cases:**
  - Borderline digits like **'1' vs '7'** or **'4' vs '9'**, which previously confused the model, are now **separated more distinctly** and improved ability to recognize subtle curves (important for '2', '3', '5', '6', '8').
- **Quantitative Improvement:**
  - Classifier accuracy improved by around **3–5%** compared to training directly on original 28x28 inputs and also reduced variance in predictions.

DEPLOYMENT: GRADIO APPLICATION

Why Gradio?

- Quick, interactive frontend without complex backend code.

Gradio interface Details:

- Input: Upload from test digit.
- Process:
  - Upscale with SR model.
  - Predict using CNN classifier.
- Output:
  - Show original ➔ upscaled ➔ predicted label.
- **Interface Features:**
  - Realtime display of input, output, and confidence scores.
  - Easy refresh and try-again options.
- Host **Local Server:**
  - Interface hosted on local address (127.0.0.1:7862).

RESULTS - METRICS

Evaluation Metric	MEASUREMENT/ Comment
Classification Accuracy	98.5% on super-resolved MNIST digits
Weighted avg Recall	99.62%
Weighted avg Precision	99.62%
Super-Resolution Quality	Good PSNR (Peak Signal-to-Noise Ratio) visually confirmed.
Inference Time:	<1 second end-to-end (SR + classification).

VISUAL EXAMPLES

```
app.launch()
```

```
/opt/anaconda3/envs/tf-env/lib/python3.10/site-packages/keras/src/saving/saving_lib.py:757: UserWarning: Skipping variable loading for optimizer 'rmsprop', because it has 36 vari
bles whereas the saved optimizer has 70 variables.
  saveable.load_own_variables(weights_store.get(inner_path))
* Running on local URL:  http://127.0.0.1:7862

To create a public link, set `share=True` in `launch()`.
```

MNIST Digit Recognition Demo

Select a test image to view original, upscaled, and predicted digit.

Select a Test Image

Image 44 (True: 3)

Clear

Submit

Original Image (28x28)

Upscaled Image (140x140)

Model Prediction

Predicted Digit: 3

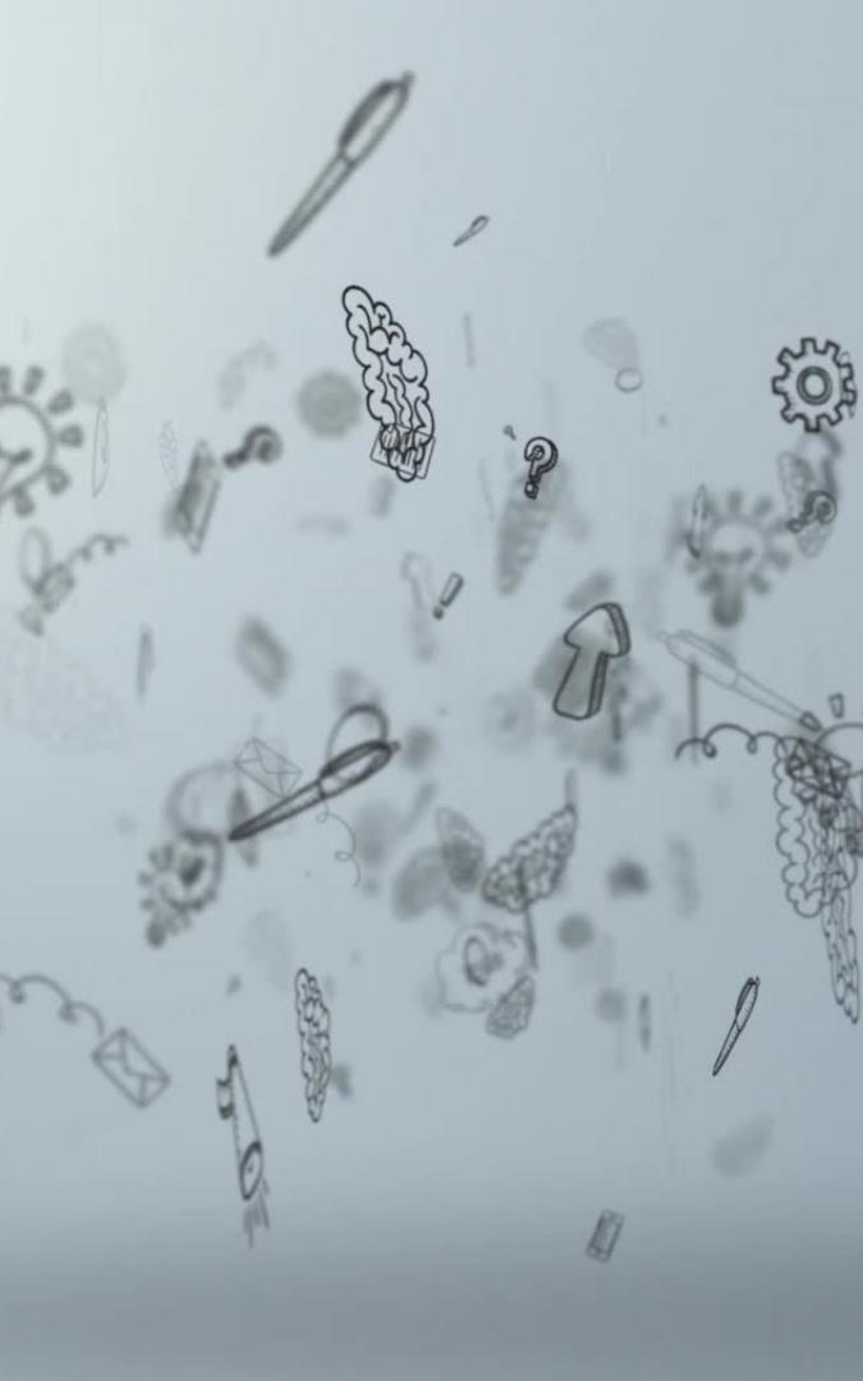
Flag

Use via API · Built with Gradio · Settings

35]:

OBSERVATIONS

- Models perform well on clean inputs.
- Slight blurring on digits with curves ('8', '9') under extreme noise.
- Residual and Dense designs improve **sharpness** and **recognition reliability** compared to naive upsampling.



FUTURE ENHANCEMENTS

- Incorporate GAN-based SR models like SRGAN.
- Add augmentation to simulate noisy/blurred real-world digits.
- Extend to larger datasets (e.g., handwritten letters, signs).
- Explore deployment on cloud or mobile apps.

CONCLUSION

- Successfully built an **end-to-end** SR + Recognition system.
- Achieved **high accuracy** and **good visual quality**.
- Demonstrated practical deployment through Gradio App.
- Future work can expand this model into real-world OCR applications.

REFERENCES

- Zhihao Wang et al., Deep Learning for Image Super-resolution, IEEE
- MNIST Dataset, Yann LeCun
- Gradio Documentation
- TensorFlow and PyTorch Libraries

ANY QUESTIONS?

THANK YOU