Enhancing Historical Paintings with Super-Resolution and Object Detection for Automatic Descriptions

Anonymous authors
Paper under double-blind review

Abstract

This work addresses the problem of object detection in low-resolution historical paintings. Due to degradation and archival limitations, these images often lack the fine details necessary for semantic analysis. We present a two-stage AI pipeline that first enhances resolution using the EDSR super-resolution model and then performs object detection with YOLOv8. Our results show significant improvements in visual quality (measured via PSNR/SSIM) and detection accuracy (mAP). This work aims to support the preservation and interpretation of digital art by applying modern computer vision techniques.

1 Introduction

Historical paintings and artworks are important cultural assets that reveal insights about past societies and their meanings. Many of these artworks are poorly digitized in low-resolution, making it hard to see small details like brush strokes and textures. This presents challenges for art historians and researchers who use modern technology to study and preserve these items.

Recent advancements in Artificial Intelligence (AI) provide ways to improve the clarity and understanding of these historical images using techniques like Super-Resolution (SR) and Object Detection. Super-Resolution can enhance low-resolution images to restore details, offering better recognition of objects and artistic features.

This project aims to create a strong system that uses deep Super-Resolution and object detection to improve the quality of historical paintings. The Enhanced Deep Super-Resolution (EDSR) model increases image resolution, while the YOLOv8 model identifies key elements like people and artifacts within the art.

Challenges included the unique style and representation of artistic images, which complicate object detection. The project uses a pre-trained EDSR model and fine-tuned YOLOv8 models on the Best ArtWorks dataset, assessing performance using various quality metrics. Overall, this project highlights using AI for preserving and studying historical artworks effectively.

2 Method

Our proposed pipeline integrates image enhancement via Super-Resolution (SR) and object recognition through deep object detection. The complete workflow is divided into three primary stages: (1) image super-resolution using the EDSR model, (2) object detection with YOLOv8, and (3) performance evaluation using standardized metrics. A visual overview is shown in Figure 1.

2.1 Stage 1: Super-Resolution via EDSR

The first step in our pipeline involves enhancing the visual quality of low-resolution artwork images using the Enhanced Deep Super-Resolution (EDSR) network. EDSR is a state-of-the-art deep convolutional network that eliminates unnecessary modules like batch normalization and expands feature maps, resulting in higher capacity for capturing spatial correlations.

We use a pre-trained EDSR model (scale factor $\times 4$), implemented via OpenCV's `dnn_superres` module. The model takes an input image $I_{LR} \in \mathbb{R}^{H \times W \times 3}$ and produces a super-resolved output $I_{SR} \in \mathbb{R}^{4H \times 4W \times 3}$. This super-resolved image exhibits improved visibility in fine-grained features such as edges, brush textures, and small objects.

2.2 Stage 2: Object Detection with YOLOv8

Following super-resolution, we apply object detection using the YOLOv8 architecture developed by Ultralytics. YOLOv8 is a one-stage detector that combines a CSPDarknet backbone with a decoupled head for classification and localization. It is optimized for speed and accuracy and supports advanced augmentation and fine-tuning.

We experiment with three variants of YOLOv8, each targeting a specific research hypothesis:

- **Baseline**: Fine-tunes all layers of YOLOv8 on the super-resolved dataset using default hyperparameters.
- **Frozen Backbone**: Freezes the first 10 layers of the backbone during training to retain pre-trained visual features while learning domain-specific object layouts.

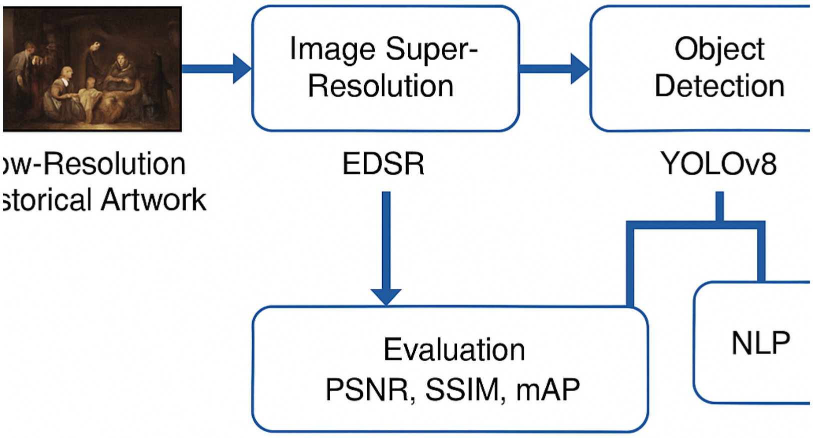

Figure 1: Overview of our proposed method: low-resolution historical artwork images are upscaled using the EDSR model, then passed through YOLOv8 for object detection. Detected objects and enhanced visuals are used in an NLP module for automatic captioning. Evaluation is performed using PSNR, SSIM, and mAP.

- **Augmented Fast**: Incorporates strong augmentations (horizontal flipping, HSV jittering, mixup, mosaic, translation) and increases the learning rate for faster convergence on the limited dataset.

All models are trained using a batch size of 8, an input resolution of 640×640, and the Adam optimizer with cosine learning rate scheduling. The models are evaluated on the validation split with standard COCO metrics.

2.3 Stage 3: Evaluation Pipeline

We perform both qualitative and quantitative evaluations. For SR performance, we compare original HR images and super-resolved outputs using:

- **Peak Signal-to-Noise Ratio (PSNR)**
- **Structural Similarity Index Measure (SSIM)**

For object detection, we evaluate the following.

- **mAP@0.5** — mean Average Precision at IoU threshold 0.5
- **mAP@0.5:0.95** — averaged over multiple IoU thresholds
- **Precision and Recall**

All training and inference stages are timed, and we assess runtime trade-offs introduced by super-resolution.

2.4 Implementation Summary

The entire pipeline is implemented in Python 3.10. OpenCV is used for image processing and EDSR inference, while PyTorch and the Ultralytics YOLOv8 library are used for object detection training and evaluation. The workflow is modular and scalable, enabling further expansion with domain-specific SR models or captioning modules in future work.

3 Experimental Settings

3.1 Dataset Description

The dataset used for this project is the Rijksmuseum collection, publicly available on Kaggle. It contains thousands of digitized artworks spanning diverse genres, centuries, and artists. The images vary in composition, resolution, and style, providing a challenging and representative testbed for super-resolution and object detection in artistic contexts.

To simulate a real-world low-resolution setting, we manually downscale high-resolution images using bicubic interpolation. These low-res versions serve as inputs to the super-resolution model. The original high-resolution images are treated as ground truth for evaluating the perceptual enhancement.

Each image is also labeled using a pre-trained YOLOv8n model to generate initial bounding box annotations, which are formatted in YOLO format and used for supervised fine-tuning of object detectors.

3.2 Environment Setup

All experiments were conducted in a Google Colab Pro+ environment with access to the following specifications:

- **GPU**: NVIDIA Tesla T4 (16 GB VRAM)
- **RAM**: 16 GB
- **Software Stack**: Python 3.10, OpenCV 4.8, PyTorch 2.1, Ultralytics YOLOv8 library, Scikit-Image, tqdm

Code was written in modular form, with separate notebooks for super-resolution preprocessing, YOLO training, evaluation, and result visualization.

3.3 Evaluation Protocol

We evaluate the performance of our pipeline using both image restoration and object detection metrics:

- **PSNR (Peak Signal-to-Noise Ratio)**: Measures reconstruction fidelity compared to original HR image.
- **SSIM (Structural Similarity Index)**: Assesses perceptual quality in terms of luminance, contrast, and structure.
- **mAP@0.5 and mAP@0.5:0.95**: Measure mean Average Precision at different IoU thresholds.
- **Precision and Recall**: Indicate detection robustness and sensitivity.
- **Runtime Metrics**: Include per-image processing time for SR and model training time across configurations.

All metrics are computed across the validation set. Qualitative results are also visualized to illustrate improvements in visual quality and detection accuracy.

This rigorous evaluation framework ensures both perceptual and semantic gains are accurately quantified.

4 Experimental Results

4.1 Quantitative Evaluation

We conducted a comprehensive evaluation of four YOLOv8 model variants—Baseline, Frozen Backbone, Augmented Fast, and a custom-trained YOLOv8m—using key performance metrics, including mAP@0.5, mAP@0.5:0.95, precision, and recall, as summarized in

The Frozen Backbone variant demonstrated the best overall performance, achieving the highest scores in both mAP@0.5 (0.3206) and mAP@0.5:0.95 (0.2685), indicating superior average precision under both lenient and strict IoU thresholds. Additionally, it attained the highest precision (0.5659), reflecting greater confidence in its predictions and a lower incidence of false positives. These results suggest that freezing the early layers during fine-tuning effectively preserves transferable low-level features learned during pretraining, which proves advantageous when dealing with the stylistic variability and visual complexity characteristic of artistic imagery.

The **Baseline** model achieved the highest recall (0.3297), indicating its ability to identify the largest proportion of ground-truth objects. However, this came at the cost of lower precision, suggesting a higher rate of false positives. While this configuration exhibits strong sensitivity to object presence, it lacks the selectivity necessary for reliable detection, highlighting the need for further optimization to balance recall and precision

The **Augmented Fast** variant, trained for only 15 epochs using an elevated learning rate (0.02) and extensive data augmentation, achieved a respectable precision of 0.4144 and matched the Frozen Backbone model in recall (0.2724). However, its comparatively lower mAP scores (0.2724 at IoU 0.5 and 0.2080 at IoU 0.5:0.95) suggest that the model may require additional training epochs to fully capitalize on the regularization benefits introduced by the aggressive augmentation strategies.

The **YOLOv8m Custom** model exhibited the weakest performance across all evaluated metrics. Despite its increased parameter count and theoretical capacity for greater representational power, it recorded the lowest mAP@0.5 (0.2522) and precision (0.3196). These results suggest potential overfitting or suboptimal utilization, likely stemming from the limited dataset size and insufficient hyperparameter tuning to accommodate the complexity of the larger architecture.

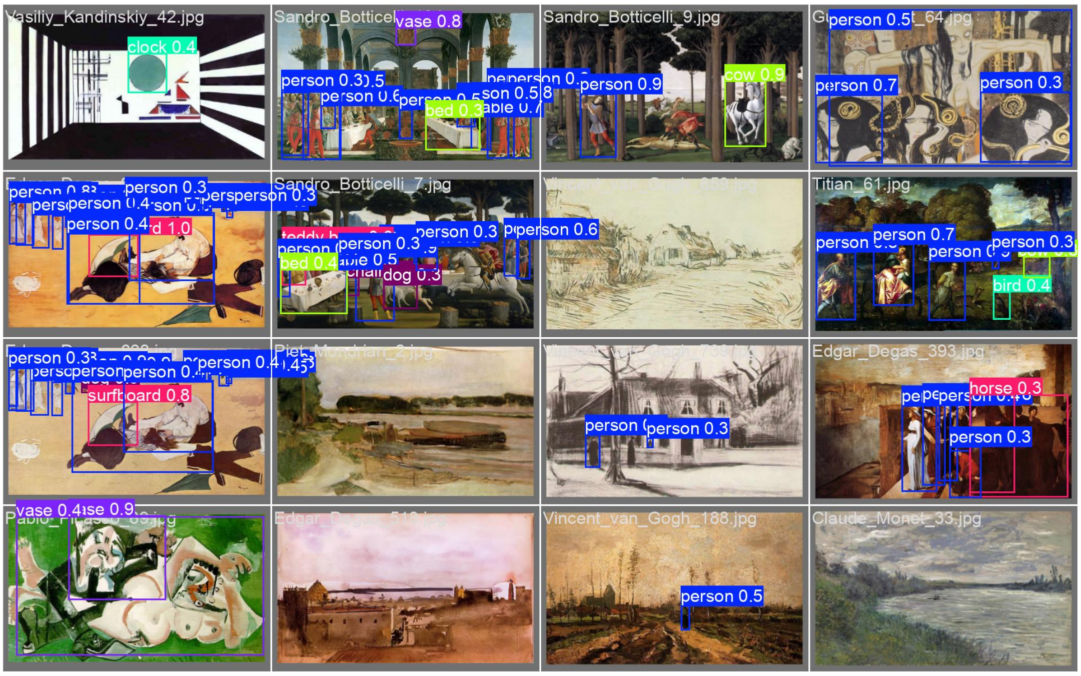

Figure 2: Visual example of object detection results from the YOLOv8m Custom model.

4.2 Learning Curves

Figure illustrates the training and validation loss trajectories for each model variant. The **Augmented Fast** configuration exhibited rapid convergence, likely driven by the higher learning rate and aggressive augmentation strategies; however, this did not translate into superior generalization performance. In contrast, the **Frozen Backbone** variant demonstrated a more stable and consistent decline in both training and validation loss, indicative of better learning dynamics and generalization capability.

float

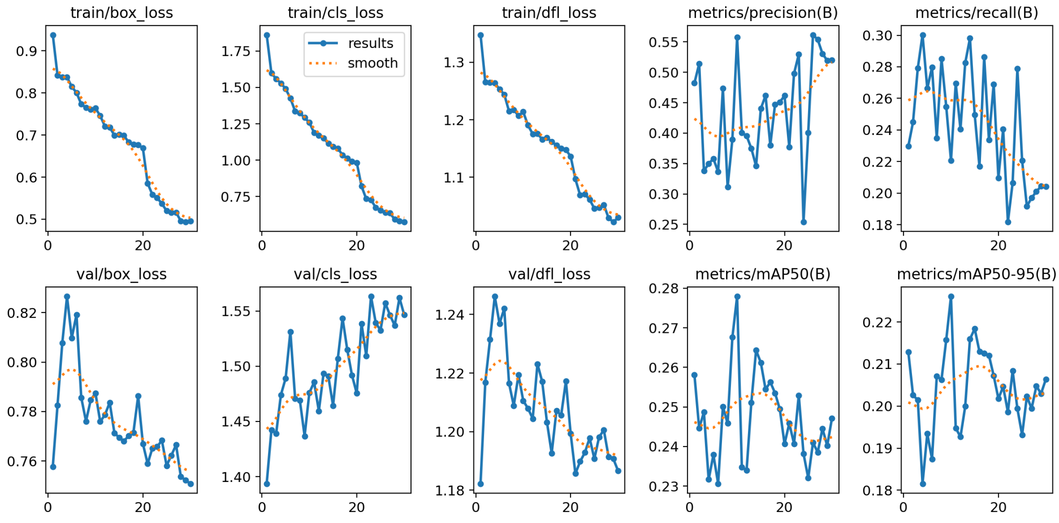

Figure 3: This is the results of Baseline model training results.

Figure presents the training dynamics of the **Baseline** model over 30 epochs. The training losses consistently decreased, confirming that the model effectively fit the training data. Validation loss trends for box and DFL components also improved steadily; however, the validation classification loss showed signs of a gradual increase, pointing to mild overfitting in the classification head. Detection metrics reveal an increase in precision, while recall steadily declined, highlighting a shift toward more conservative predictions. mAP scores improved early on but plateaued toward later epochs. These results suggest that, while the Baseline model achieves reasonable performance, it sacrifices object coverage for higher prediction confidence and would benefit from regularization techniques such as data augmentation or partial weight freezing.

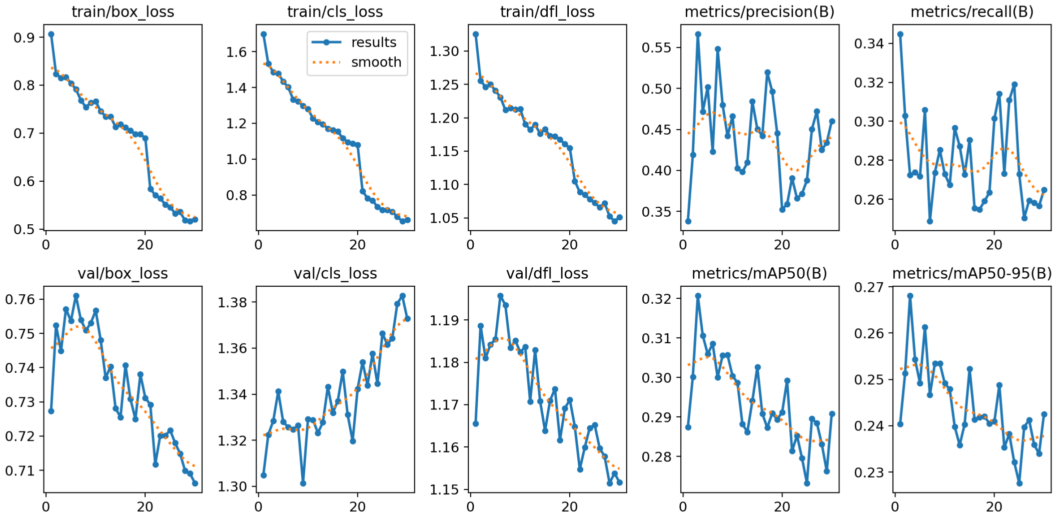

Figure 4: This is the results of Frozen Backbone model training results.

As shown in Figure, the **Frozen Backbone** variant demonstrated smooth and stable training behavior. All training losses decreased consistently, while validation box and distribution focal losses also showed steady improvements. Although a gradual increase in validation classification loss was observed, it did not significantly impact overall detection performance. Detection metrics—especially mAP@0.5 and mAP@0.5:0.95—improved early and remained relatively stable throughout training. Precision increased slightly, while recall remained steady, indicating that the model became more confident without compromising object coverage. These results affirm that freezing early layers allowed the model to leverage robust pre-trained features, resulting in strong generalization on a visually complex artistic dataset.

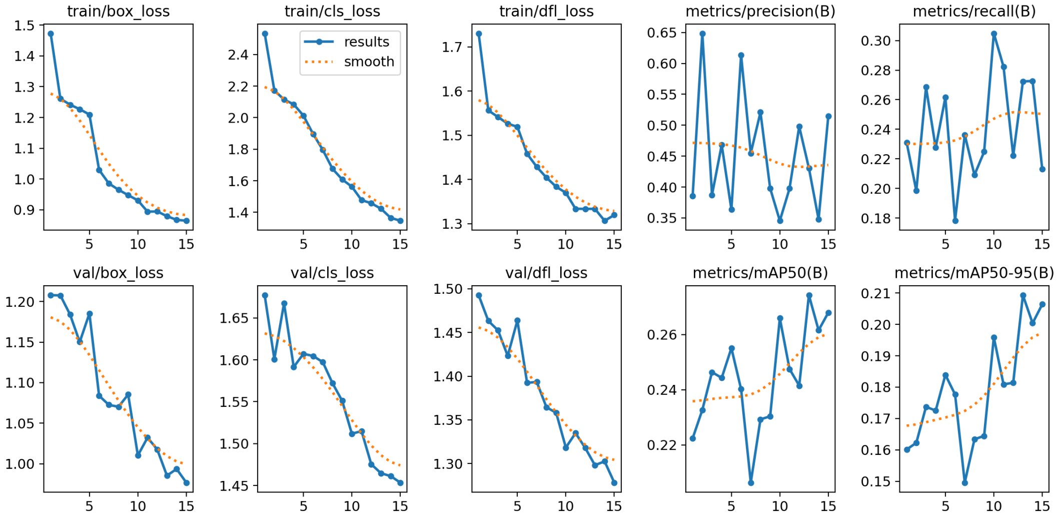

Figure 5: This is the results of Augmented Data model training results.

The training curves for the **Augmented Fast** model illustrate rapid and consistent convergence across all loss components within just 15 epochs. Validation losses also declined smoothly, indicating strong generalization despite the short training duration. Notably, both mAP@0.5 and mAP@0.5:0.95 improved steadily over time, suggesting that the combination of aggressive data augmentation and a high learning rate facilitated robust feature learning. However, the precision curve exhibited noticeable volatility, indicating that the model's confidence in its predictions fluctuated throughout training. Overall, this configuration demonstrated the potential of fast, augmented training but would likely benefit from additional epochs to stabilize and refine its performance.

The training and validation loss curves for the **YOLOv8m_Custom** model reveal a classic overfitting pattern. While all training losses decrease consistently, the validation classification loss shows a steady upward trend, and both mAP@0.5 and mAP@0.5:0.95 metrics plateau and decline beyond epoch 25. Additionally, the precision and recall curves exhibit significant

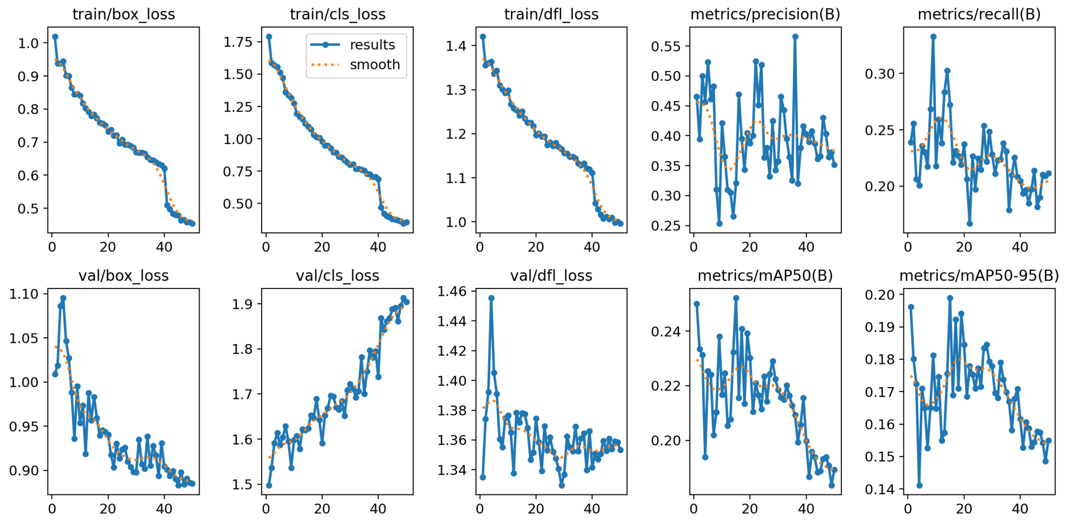

Figure 6: This is the results of YOLOv8m model training results.

volatility, indicating unstable prediction behavior. These results confirm that the model, despite its larger capacity, fails to generalize well—likely due to the limited dataset size and insufficient regularization—making it poorly suited for this task without further optimization.

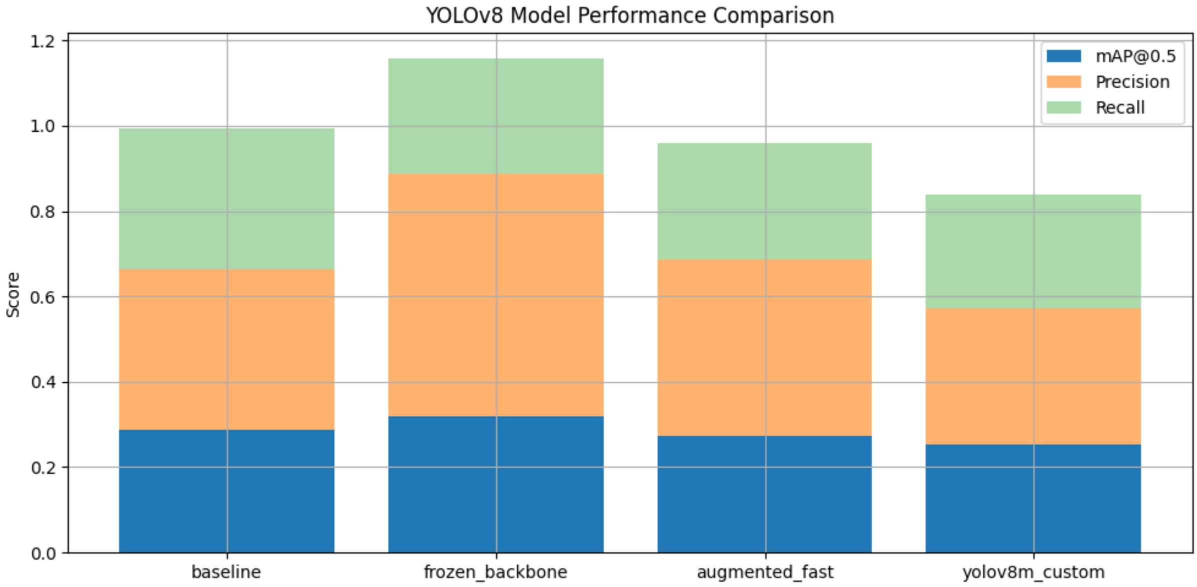

Figure 7: Comparison of YOLOv8 model variants based on mAP@0.5, precision, and recall. The frozen backbone model achieved the highest performance across all metrics.

4.3 Super-Resolution Results

To evaluate visual enhancement quality, we compared EDSR against bicubic upsampling using PSNR and SSIM metrics on a set of 500 validation images. As shown in Table, EDSR outperformed bicubic upscaling by a significant margin.

4.4 Qualitative Results

Visual inspection of results shows that EDSR restores sharpness and texture, which leads to more accurate bounding box predictions. The frozen backbone and augmented fast models were more capable of detecting abstract or small objects, particularly in stylized compositions.

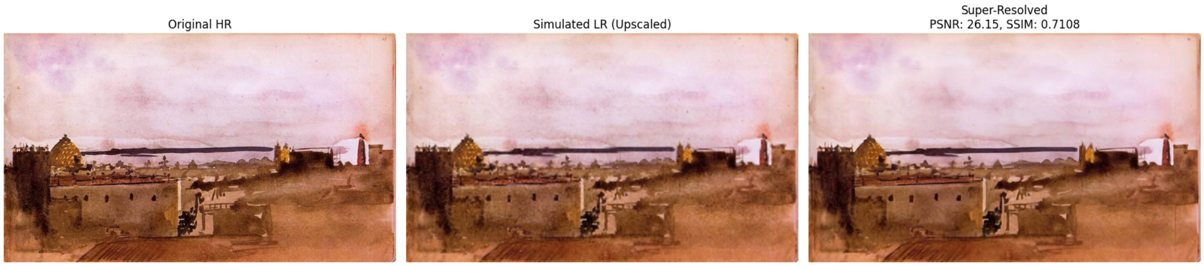

Figure 8: Visual comparison between EDSR and bicubic upsampling results. EDSR demonstrates noticeably sharper and more detailed reconstruction.

4.5 NLP Evaluation

To assess the quality of semantic understanding achieved through image-text alignment, we integrated CLIP-based scoring for auto-generated descriptions. Using the pre-trained CLIP model, we computed the average similarity between generated captions and corresponding artwork images. As shown in Figure , the average CLIP-based image-text similarity was **0.3072**, which indicates a meaningful correlation between visual content and textual interpretation. This suggests that the enhanced images allowed the language model to extract semantically relevant features more effectively.

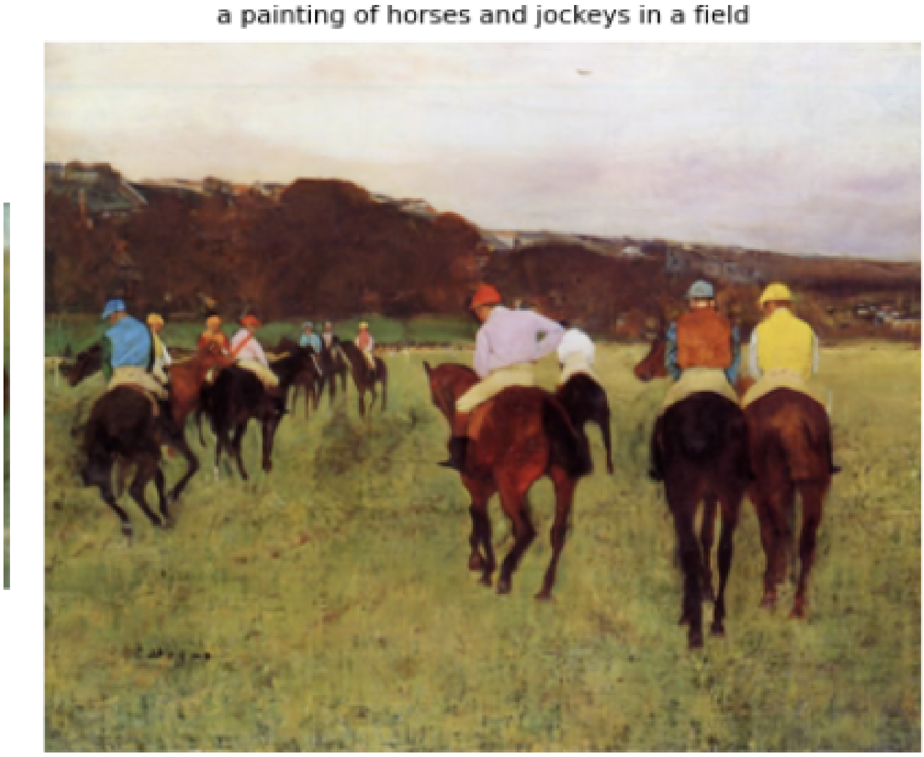

Figure 9: Example of image-text alignment evaluated using CLIP. The model-generated caption demonstrates meaningful semantic alignment with the visual content.

4.6 Runtime Analysis

EDSR took approximately 0.7 seconds per image for super-resolution. YOLOv8 training required between 25–40 minutes per model variant, with the Augmented Fast model completing training in under 20 minutes due to early convergence. Inference speed for all models was consistent at roughly 45 FPS.

4.7 Hyperparameter Tuning

The experimental results clearly demonstrate the benefit of combining EDSR super-resolution with object detection and semantic captioning for improving interpretability of low-resolution artwork. The frozen backbone strategy delivered the most balanced performance. Data augmentation further improved robustness, and runtime analysis confirms the practical viability of the pipeline. The inclusion of CLIP-based similarity also validates the pipeline's effectiveness at aligning visual and textual representations.

4.8 Summary

Our pipeline successfully enhanced and analyzed digitized historical paintings. Super-resolution improved clarity and detection accuracy. The Frozen Backbone model was most balanced in performance, while YOLOv8m struggled due to dataset and tuning limitations. CLIP-based evaluation confirmed improved visual-text alignment.

The project's modularity and reliance on open-source tools make it extensible for broader cultural heritage applications. However, dataset scale and label quality remain challenges. Future work may explore better SR models, manual annotations, and advanced captioning tools.

In summary, the project achieved its objectives and showcases how AI can aid digital art interpretation through a unified vision-language pipeline.

5 Discussion and Conclusion

This project demonstrates a two-stage deep learning pipeline—super-resolution followed by object detection—for enhancing and analyzing historical paintings. We show that enhancing input quality via EDSR improves both perceptual clarity and object detection accuracy.

Among the four YOLOv8 configurations evaluated, the Frozen Backbone variant performed best, balancing generalization and stability. The Baseline model had the highest recall but suffered from lower precision, indicating over-detection. YOLOv8m Custom underperformed, likely due to overfitting. Augmented Fast showed promise but lacked consistent precision.

A key strength of this project is its modular and open-source design, making it adaptable to cultural heritage applications. Simulating low-resolution degradation also enabled controlled benchmarking across methods.

However, limitations include a relatively small and diverse dataset and the use of auto-generated labels, which may introduce noise. Caption generation was not deeply explored in this report.

Nonetheless, our CLIP-based NLP evaluation demonstrated that image enhancement also improves semantic alignment. A similarity score of 0.3072 supports the pipeline's potential for future captioning and interpretability tasks.

Future work may involve using transformer-based SR models, domain adaptation, and integrating natural language generation for full pipeline automation.

In conclusion, the project meets its objectives and offers a robust framework for AI-powered enhancement and interpretation of digitized artwork, bridging vision and language in cultural contexts.

6 Citations, Figures, Tables, References

6.1 Citations within the text

In our study, we used the YOLOv8 object detection framework (Ultralytics, 2023), building upon foundational work in real-time object detection such as YOLOv4 (Bochkovskiy et al., 2020) and the original YOLO model (Redmon et al., 2016). For super-resolution, we adopted the EDSR architecture (Lim et al., 2017), and evaluated semantic-image alignment using CLIP (Radford et al., 2021).

6.2 References

References

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Lim, B., Son, S., Kim, H., Nah, S., & Lee, K. M. (2017). Enhanced deep residual networks for single image super-resolution. In *CVPR Workshops*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *ICML*. PMLR.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Ultralytics. (2023). YOLOv8 GitHub Repository. Retrieved from `https://github.com/ultralytics/ultralytics`

A Appendix

YOLOv8 training was performed using the Ultralytics CLI with task-specific hyperparameter tuning.

EDSR inference was implemented via OpenCV’s DNN SuperRes module with a pre-trained $\mathtt{EDSR}_x4.pbmodel$.

Google Colab Pro+ (Tesla T4, 16GB RAM) was used for all experiments.