A Hybrid Approach of EDVR and YOLOv8 for Super-Resolution

Abul Monsur Mannan
Department of Computing and Software Engineering
Athlone Institute of Technology (AIT)
Athlone, Ireland
A00304503@student.ait.ie

***Abstract*— Advances in super-resolution (SR) and object detection have often been used largely in parallel within the domain of computer vision. While Enhanced Deformable Video Restoration (EDVR) by Wang et al. [8] excels in generating high-resolution images, models like YOLOv8 by (Ultralytics, 2023)[1] are optimized for detecting and segmenting objects within those images. This research seeks to synergize these specialized capabilities into a unified framework while adopting a 'qualitative methodology' approach. Utilising a PyTorch-based implementation, the study fuses the aligned feature maps produced by EDVR with YOLOv8-seg's contextual layers on a standard Set5 dataset. The hypothesis posits that the fusion of features from both models will result in high-quality super-resolution outputs, augmented with precise object detection and segmentation. The proposed model has applications across various domains, including surveillance, medical imaging, and autonomous vehicles.**

I. Introduction

In the context of computer vision, Super-Resolution (SR) is a foundational technique for enhancing image and video frame quality. Notable progress has been made by models like Enhanced Deformable Video Restoration (EDVR) [8] in achieving Super Resolution. However, the integration of object detection and segmentation into SR remains an active and exploratory field. Models such as YOLOv8-seg [4] have shown strong segmentation abilities, particularly for capturing contextual information in images.

A research gap exists where EDVR excels in video super-resolution and YOLOv8-seg specializes in object detection and segmentation. There's a lack of methodologies that combine their strengths to comprehensively enhance super-resolution quality in images and video frames.

This study's primary objective is to propose an innovative model that merges feature maps from YOLOv8-seg with the aligned features of EDVR. The goal is to achieve high-quality super-resolution while preserving accurate object detection and segmentation. The qualitative approach employs PyTorch in Python, using Jupyter Notebooks, VSCode, and GPU acceleration via Google Colab. Datasets like VID4, Set5, Set14, and potentially from the REDS dataset are used for empirical validation.

The research hypothesis suggests that fusing feature maps from YOLOv8-seg and EDVR will generate high-quality image and video frames with improved perceptual quality and accurate object detection masks. This hypothesis draws inspiration from works like Fast-SAM [26], SEEM [21], and Improved EDVR [20].

The study's significance lies in its potential contribution to computer vision, particularly in tasks requiring super-resolution alongside object detection and segmentation. This impact spans domains such as surveillance, medical imaging, and autonomous driving systems. The study bridges a gap in the current understanding, opening avenues for advanced applications in various sectors.

II. Literature Review

Researchers have introduced diverse learning models to tackle the super-resolution challenge, each with unique strengths and limitations. Interpolation, a widely used technique for resolution enhancement, is countered by methods like sparse coding and dictionary learning to map low-resolution images to high-resolution space. Patch-based methods like self-model and non-local means enhance textures and details but might introduce blocking artefacts. Deep learning models, particularly Convolutional Neural Networks (CNNs), such as SRCNN[9], VDSR[27], and EDSR [6], have excelled in capturing nonlinear LR-HR relationships, offering promising results in super-resolution. GAN-based models like SRGAN and video-specific models like EDVR show potential for high-quality outputs.

In the realm of video super-resolution (VSR), techniques using deep neural networks have advanced significantly. Spatial and temporal information play pivotal roles, with models like DUF [28] and EDVR incorporating optical flow and temporal dependencies to enhance quality. Spatial information encompasses pixel values, gradients, and structures, while temporal information addresses changes over time in sequences. Techniques like patch-based methods, CNNs, sparse coding, and motion compensation extract spatial details, while optical flow estimation and temporal super-resolution networks capture temporal changes.

Comparing super-resolution methods reveals their strengths and weaknesses. Bicubic interpolation results in blurry images, while deep learning models like CNNs capture complex relationships but require substantial resources. Proper dataset selection and augmentation are essential for robust performance. Two models, EDVR and YOLOv8-seg, are chosen for research. EDVR excels in video super-resolution, leveraging temporal coherence and motion compensation, yet faces complexity challenges. YOLOv8-seg combines object detection and semantic segmentation for robust performance but may be impacted by detection and segmentation inaccuracies. Both models guide the research towards effective super-resolution solutions.

III. PROBLEM STATEMENT

The focus of this survey is twofold. First, a basic survey of SR models and the selection of base models for the second step of this study. The second part is to develop image resolution models that can effectively improve the quality and analysis of images, in computer vision applications.

A. Challenges and Limitations of Traditional Approaches

Conventional methods for enhancing image resolution often use interpolation techniques like interpolation to enlarge low-resolution images (Guo et.al., 2021). However, these methods are limited in their ability to capture the details and textures found in high-resolution images. Furthermore, they struggle with handling image structures such as edges and textures leading to unrealistic results (Guo et al., 2021). Moreover, these approaches are not well suited for enhancing video sequences as they do not effectively utilize the information that's essential for accurate resolution enhancement.

The survey's primary objective is the exploration of image resolution enhancement models to improve image quality and analysis in computer vision applications. The focus of the survey is to address limitations of traditional methods through advanced techniques like YOLOv8-seg and EDVR. It is found that traditional approaches, which rely on interpolation fail to capture high-resolution details, struggle with image structures, and are inadequate for video sequences. YOLOv8-seg excels in object detection and segmentation that leverages contextual information for effective super-resolution and maintains object boundaries in low-resolution images. It demonstrates strength to handle video sequences along with facing challenges in accuracy and object occlusions, alongside computational complexity. On the other hand, EDVR specializes in enhancing video resolution by the use of temporal coherence between frames and motion compensation that offer high-quality, realistic results. Nonetheless, EDVR counters weaknesses such as computational intensity, data diversity dependence, artefact amplification, motion complexities, and overfitting risks. Hence, these models show promise to enhance super-resolution efficiency, but their strengths and limitations warrant attention for future research refinement.

IV. RESEARCH OBJECTIVES

The primary aim of the study is to conduct a survey on learning models used for achieving super-resolution (SR) in images. The study is motivated to explore the advancements and techniques in learning models that can enhance image resolution and quality. The study will examine the techniques employed in these models, how the contributors evaluate their performance and identify their limitations. Through this analysis, the aim is to gain insights into the strengths, weaknesses, and areas for improvement of models.

In addition, the research aims to propose, implement, and evaluate novel image resolution model. The study will also explore techniques and existing models that can enhance image quality through SR. The focus will be on developing efficient models, with improved generalization and robustness capabilities. As well as, the study intends to undertake an investigation into the possibility of boosting video SR performance by using two models which are possibly YOLOv8-seg and EDVR.

This study shall, therefore, involve the extraction and integration of the different video frames with the aim of discovering the way these models can complement each other and can be used for video SR.

V. RESEARCH METHODOLOGY

The research methodology is composed of a comprehensive investigation of techniques to enhance image and video resolutions. The focus of the study is on both classical algorithms and deep learning-based methods.

A. Methodology breakdown

In this study, we are following a qualitative research approach.

The first part goes with the investigation of the methods, SR models and techniques followed by the authors of YOLOv8 and EDVR to fulfil our research objective. Following this, we examine the architecture of SEEM model from (Lu et al.)[40] and Improved-EDVR from Huang et al. [38] to materialize our research objective. These papers and their mentioned architecture and models are peer-reviewed in several other papers and practically implemented in real-life applications.

The second part consists of an in-depth examination of the methods and techniques the authors followed in those models to achieve video/image SR.

In the last and third part, we provide a detailed explanation of the algorithms, frameworks, and tools we utilize for implementing and testing the our proposed models. The model creation, implementation and results will come in the following section of this paper.

B. About EDVR and YOLOv8

This study focuses on evaluating YOLOv8 and EDVR, contributing to the existing knowledge of image/video super-resolution. The EDVR architecture comprises components such as Pyramid, Cascading, and Deformable (PCD) alignment module for large motion handling, Temporal and Spatial Attention (TSA) fusion module for feature emphasis, Feature extraction network, Enhanced Deformable Convolutional Network (EDCN) for restoration, and modules for Reconstruction and PreDeblur. The latter mitigates blur in video sequences by learning complex mappings. YOLOv8-seg utilizes CSPDarknet53 backbone with Cross-Stage Partial (CSP) connections for feature extraction. Beginning with convolutional and pooling layers, CSPDarknet53 captures low-level to abstract features, enabling effective representation. Extracted features serve object detection and semantic segmentation tasks, achieving a balance between fine details and global context through well-designed architecture and connections.

C. Implementation background

To facilitate our work we have inspired by and chosen two models for achieving a better image/video SR for our study. These models are SAM-guided Ed refinement Module (SEEM) by (Lu et al., 2023) and Improved EDVR by (Huang & Chen, 2022). We follow their method of work and tools to work with EDVR and YOLOv80seg pretrain models.

- SEEM:

SEEM's plug-in module boosts the EDVR's performance and quality by using the semantic information from SAM to improve the alignment and fusion of multiple frames. According to the paper, SEEM can enhance both the foreground and background regions of the video frames, resulting in more accurate and realistic super-resolution outputs. For example, in section 4.2 of the paper, the authors show some visual comparisons between EDVR and EDVR+SEEM on the REDS dataset. They demonstrate that SEEM can better handle large motions, occlusions, and complex textures, such as the moving car, the occluded person, and the brick wall.

They also provide some quantitative results in Table 2 [40] of the paper, where they report that SEEM can improve the PSNR and SSIM metrics of EDVR by 0.14dB and 0.003, respectively, on the REDS dataset.

Moreover, SEEM can also reduce the number of parameters and FLOPs of EDVR by 11.6% and 10.8%, respectively, without sacrificing performance, as shown in Table 3 [40] of the paper. Therefore, SEEM's plug-in module can boost EDVR's performance and quality by utilizing a more robust and semantic-aware prior for VSR.

- IMPROVED EDVR:

The authors identify two main challenges in VSR: accuracy and the need for high-speed, possibly real-time, models.

To address these challenges, the authors propose several improvements – 1) a preprocessing module, 2) A temporal 3D, 3) a convolutional fusion module, and 4) a new reconstruction block. Additionally, the authors employ multiple programmatic methods to accelerate both the model training and inference processes, making the model more practical for real-world applications.

We assume, the proposed model not only outperforms the baseline EDVR model in terms of PSNR and SSIM but also maintains a similar level of parameter count. This makes the model both robust and efficient, reducing timing cost and memory consumption while delivering extraordinary performance.

- PROGRAMMING TOOLS AND FRAMEWORKS

Programming tools: PyTorch - In the context of computer vision, PyTorch provides a versatile and flexible platform for developing and deploying deep learning models.

Platform: Google-Colab and Amazon's AWS platforms give some sort of free services.

- CONCLUSION

Video Super-Resolution (VSR) is a growing field in computer vision aimed at enhancing video sequence resolution. The foundational Enhanced Deformable Video Restoration (EDVR) model employs deformable convolutions and attention mechanisms to improve VSR. An advanced EDVR version introduces innovations like Temporal 3D Fusion and a new Reconstruction Block, yielding improved performance metrics and parameter efficiency. Another model, using the Segment Anything Model (SAM), introduces SAM-guided Ed refinement Module (SEEM) to enhance alignment and fusion in VSR. These models showcase evolving VSR methods to address accuracy and efficiency challenges, while acknowledging the study's qualitative nature and reliance on select peer-reviewed papers for validation in the dynamic machine-learning landscape.

VI. THE PROPOSED MODEL

We already come across that deformable convolutional networks (DCNs) improves object detection and semantic segmentation task. We also found that SAM's segmentation task gave a tremendous boost in various tasks in the computer vision field. The super popular and practical model YOLOv8 showed us how SAM's image segmentation capability helps to achieve better results. Also, the plug-in module of SEEM showed a use-case of SAM's extracted feature maps in various ways which boost the VSR.

All of these models and techniques guide us to come up with an idea to improve the super-resolution of image and video frames.

A. Our module hypothesis:

To formulate a model, we intend to use the YOLOv8-seg model's feature maps and fuse those features with the EDVR's aligned feature. We use the following approach to achieve our goal:

- First, we aim to collect all the feature maps from the image/video frame produced by the YOLOv8-seg model.
- This will give us an idea of where the objects are located in each frame through feature masks. This gives us the internal information of the feature maps like shape and size.
- Next, we can use EDVR to super-resolve each frame of the video. This will give us high-quality image frames with improved perceptual quality. Tweaking EDVR architecture will be required here.
- Then, we can fuse the features extracted from YOLOv8-seg with the aligned features from EDVR to generate high-quality image or video frames with very good image segmentation.

We hypothesise that the above approach allows us to generate high-quality image/video frames with object detection masks. However, it is important to note that this approach may require significant computational resources due to the complexity of both models.

- Tools: We, initially, use Jupyter Notebook and VSCode as IDE to investigate, debug and create our module's code. After that, we transfer our model to Google-Colab's GPU for faster output.
- Dataset: In our task, we aim to use a small dataset of image frames from VID4, Set5 or Set14 and possibly from REDS dataset. We aim to carry on our tasks on a single image first then go on with a collection of video image frames.
  - About Result: As this study is qualitative, our primary aim to produce results from this proposed model are images. Then we compare those with the input images and ground truth images provided by the well-known image frames of our selected datasets.

So, qualitative visual comparisons between different resolution models are important part of this study.

B. *Second part*

Our secondary aim is to produce mathematical measuring techniques like PSNR and SSIM if we can have enough time. There has been always a time constraint in this type of study to get a full-fledged outcome.

The feature maps generated from the YOLOv8 model will be concatenated into the PCD alignment module or/and the TSA fusion module. The end result will be obtained from the EDVR model.

We experiment this by writing codes using Pytorch on the local machine with CPU and with single GPU from Google's Colab environment. Our initial aim is to set up a workable test and train environment and then develop code to make our proposed model fit in the EDVR model. In this case, the main architecture file of the EDVR model will be customized.

VII. Experiments, Results and discussion

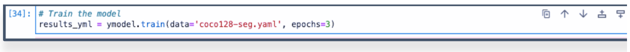
A. *YOLOv8-seg uses:*

Initially, we start to implement the YOLOv8-seg pre-trained model to obtain and examine the feature mask from the 3 outputs of the final 3 c2f modules. All the c2f module are 3x3 conv.

We use Jupyter Lab and VSCode on local machine. We found the YOLOv8 document is clear enough to carry out this task. The creators and maintainers (Ultralytics, 2023) make this easy for everybody.

We use instance segmentation from the pre-trained model (yolov8s-seg) for our task*[2]*.

But, to better understand and tuning the hyperparameter, we also used the YOLOv8-seg.yml file like this "**model = YOLO('yolov8n-seg.yaml')**". Then we trained it on COCO128 dataset like this:



```
[34]: # Train the model
      results_yml = ymodel.train(data='coco128-seg.yaml', epochs=3)
```

Figure: Training with YOLOv8-seg.yml

Some results from these are as follows:

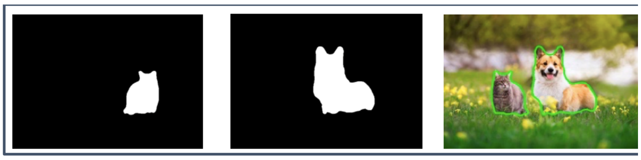
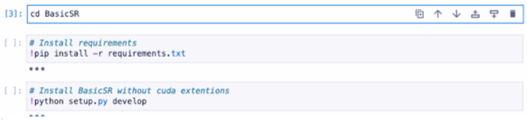


Figure: Feature masks from YOLOv8-seg

B. *EDVR uses:*

Because of the old and complex documentation, implementing EDVR model was tedious. For investigating the model, the simplest way to use the model are as follows:

1. Install and update basic environmental packages from Python and Pytorch's torch and torchvison.
2. Clone the latest EDVR packages from BasicSR found in Github repository (Xinntao, *Xinntao/EDVR: EDVR has been merged into BASICSR.*) [25]
3. The following should be done in order to install EDVR model as shown in this image:



```
[3]: cd BasicSR

[ ]: # Install requirements
     !pip install -r requirements.txt
     ...

[ ]: # Install BasicSR without cuda extentions
     !python setup.py develop
     ...
```

Figure: EVDR installation after cloning the EDVR's GitHub repository

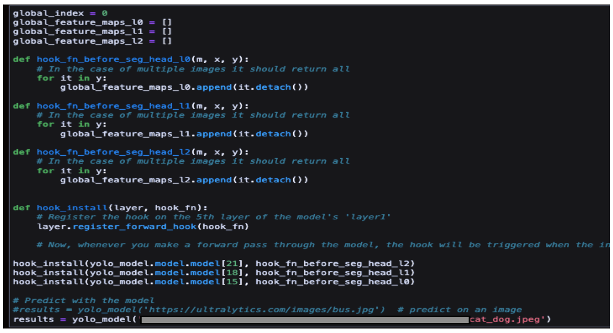
4. Train and test can be done by keeping the datasets on disk which is the most easiest way or can be done other ways mentioned in their Github repo (link given earlier)
5. We uses image frames from the dataset REDS and VID4. Training and testing required greater computational power and time on CPU. Even with single GPU, it could require hours of time. We use Google's Colab environment.
6. The results are impressive as stated in the original paper. Some results are here which are found same in our experiments*[29]*.

C. *Developing our proposed module:*

- Feature extraction from YOLOv8:

As we plan, according to our proposed method, we need to gather feature maps from the YOLOv8-seg model. We used a pretrained model for this purpose. In this case, it is straightforward as no configuration is needed or intended for our study.

We write a block of code using Pytorch to extract feature maps which are generated from the three c2f module.

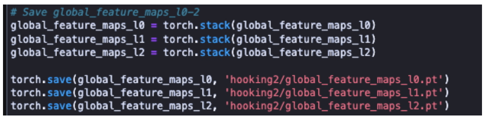


```
global_index = 0
global_feature_maps_l0 = []
global_feature_maps_l1 = []
global_feature_maps_l2 = []

def hook_fn_before_seg_head_l0(m, x, y):
    # In the case of multiple images it should return all
    for it in y:
        global_feature_maps_l0.append(it.detach())

def hook_fn_before_seg_head_l1(m, x, y):
    # In the case of multiple images it should return all
    for it in y:
        global_feature_maps_l1.append(it.detach())

def hook_fn_before_seg_head_l2(m, x, y):
    # In the case of multiple images it should return all
    for it in y:
        global_feature_maps_l2.append(it.detach())


def hook_install(layer, hook_fn):
    # Register the hook on the 5th layer of the model's 'layer1'
    layer.register_forward_hook(hook_fn)

    # Now, whenever you make a forward pass through the model, the hook will be triggered when the in

hook_install(yolo_model.model.model[21], hook_fn_before_seg_head_l2)
hook_install(yolo_model.model.model[18], hook_fn_before_seg_head_l1)
hook_install(yolo_model.model.model[15], hook_fn_before_seg_head_l0)

# Predict with the model
#results = yolo_model('https://ultralytics.com/images/bus.jpg')  # predict on an image
results = yolo_model('[redacted]cat_dog.jpeg')
```

Figure: Code example- Feature extraction from YOLOv8-seg model with a sample image

This test was successful. So, we happily processed to the next stage.

We did the same process to extract features by using the Pytorch's 'hook_install' method from some image frames from Set5 (from calendar) dataset. We saved our 3 feature maps from the 3 outputs (which are used later to detection module by YOLOv8) as Pytorch model checkpoint files (with .pt extension) in a separate directory as shown in the figure below:

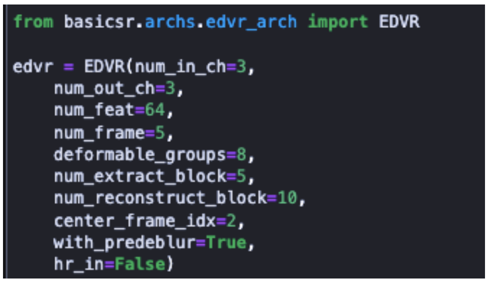


```
# Save global_feature_maps_l0-2
global_feature_maps_l0 = torch.stack(global_feature_maps_l0)
global_feature_maps_l1 = torch.stack(global_feature_maps_l1)
global_feature_maps_l2 = torch.stack(global_feature_maps_l2)

torch.save(global_feature_maps_l0, 'hooking2/global_feature_maps_l0.pt')
torch.save(global_feature_maps_l1, 'hooking2/global_feature_maps_l1.pt')
torch.save(global_feature_maps_l2, 'hooking2/global_feature_maps_l2.pt')
```

Figure: Saving feature maps from YOLOv8-seg model with a sample image

D. *Use of EDVR model on extracted feature maps from YOLOv8*

Now, we started to implement EDVR for our goal of this study. For the sake of ease, we call the 'EDVR_arch' file directly in our code like below:

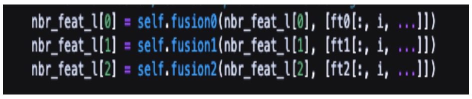


```
from basicsr.archs.edvr_arch import EDVR

edvr = EDVR(num_in_ch=3,
     num_out_ch=3,
     num_feat=64,
     num_frame=5,
     deformable_groups=8,
     num_extract_block=5,
     num_reconstruct_block=10,
     center_frame_idx=2,
     with_predeblur=True,
     hr_in=False)
```

Figure: edvr_arch call

We have tested rigorously to find out an optimum EDVR model for our use.

Now, comes our main experimental part to use the feature maps' information from YOLOv8-seg model and fuse/concat them in the EDVR model. Our experiments on tweaking the EDVR's main architecture file (edvr_arch) mostly failed except one. We have found that when PCD module forwards its outputs for feature alignment with neighbouring features is the best place to merge the feature maps gathered from YOLOv8-seg model. The code block we developed is shown in the figure below:



```
nbr_feat_l[0] = self.fusion0(nbr_feat_l[0], [ft0[:, i, ...]])
nbr_feat_l[1] = self.fusion1(nbr_feat_l[1], [ft1[:, i, ...]])
nbr_feat_l[2] = self.fusion2(nbr_feat_l[2], [ft2[:, i, ...]])
```

Figure: Fusion with the neighbouring features of EDVR model

This concept is applied to the EDSR model as well on a single image. We incorporate two fusion module to achieve our purpose:

- Attention Fusion Module: Multiplies the input tensor **x** with attention maps generated from feature tensors **fts**(feature maps).
- Stack Fusion Module: Concatenates the input tensor **x** with attention maps and then applies a 1x1 convolution.

As the module names indicates, these are designed to perform attention-based fusion of feature maps.

We know that element-wise multiplication is more appropriate for tasks where spatial focus is important, while concatenation is better for tasks requiring richer feature representations.

Both modules are designed to be initialized at runtime based on the shape of the input tensors, which offers flexibility but may not be optimal for all use-cases.

VIII. THE FINAL MODEL STRUCTURE:

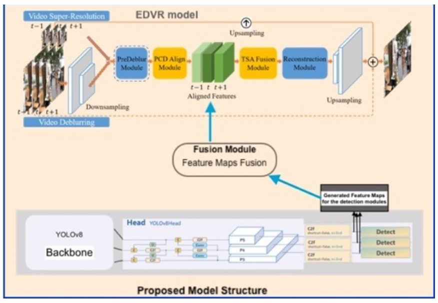


Figure: The final model architecture

E. *Reason for the fusion models*

While experimenting the feature maps from YOLOv8-seg and EDVR models, we recognize that the models are creating and working on two different types of dimensions and sizes respectively. This resulted in many considerations to develop a fusion model, like, loss of information, performance and unwanted or added information, etc.

EDVR expects input feature size containing 64 channel number of intermediate features with a minimum 5 number of frames etc. While YOLOv8's generated feature map sizes are different.

For example, if the input is with a batch of 16 frames each of size 128x128 with 3 colour channels, a feature map after a convolutional layer might have a shape like [16, 64, 128, 128]. On the other hand, with the same batch of 16 images each of size 416x416 with 3 colour channels, we might encounter feature maps with shapes like [16, 512, 13, 13], [16, 256, 26, 26], and [16, 128, 52, 52] at different scales in the network.

F. *Results and discussion*

- Result in quality:

Although we had limited resources, we have found impressive results. As mentioned earlier, we conduct qualitative research and quantitative research is not our agenda in this study.

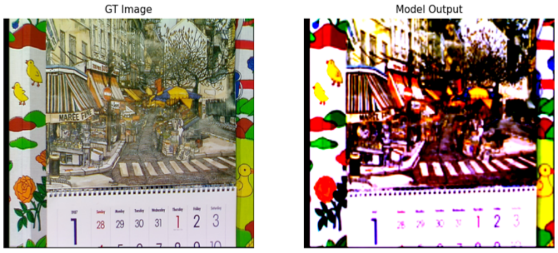


Figure: Comparing output with the ground truth calendar image frame from Set5 dataset

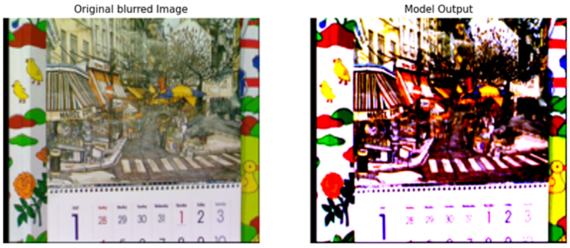


Figure: Comparing output with the blurred LR calendar image frame from Set5 dataset

Though our focus is not to compare, we can see this produces good quality of outputs that match or even produce better quality of outputs. Our model produces better quality in terms of resolution against the LR images.

The output shows some image augmentation results which can be managed via image normalization within the 'dataloader' function usage.

Also, as we have mentioned, because of the unavailability of required GPUs and time constraints, we could not carry out a full train and test of the model on a full dataset.

- PSNR result:

We have carried out a simple PSNR test on our experiment. Compared with the original paper's EDVR measures, we found a lower value. In terms of PSNR, the better value indicates a better model/test. So we can conclude that our proposed model gives poorer results than the original model.

G. Implementation problems

The original EDVR model is very resource hungry and it takes longer times with GPUs. We must confess that our experiments carried out in local machines with regular Intel-based CPUs and Google Colab's GPUs which are proved not good options for this entire task. Even a PNG formatted image with size 1280 by 720 could not be processed by the local CPU and not even Google-Colab's single GPU which provides 128MB RAM.

Overall, the architecture is a way of using two distinct computer vision models for super-resolution problems. Combining these two models on huge data of image frames will require extra computational power and resources.

IX. Conclusion

This study aimed to evaluate and propose a hybrid super-resolution model that synergizes the capabilities of YOLOv8-seg and EDVR. The primary focus was on the application of computer vision tasks, particularly in enhancing the quality of image and video frames via Super Resolution (SR) techniques.

Our proposition involved extracting feature maps from YOLOv8-seg and aligning these with features from EDVR to produce high-quality image frames with enhanced object detection masks. Despite the high computational requirements, the initial results showed promise. We introduced two fusion modules—Attention Fusion and Stack Fusion—to handle the different types of dimensions and sizes produced by each of the individual models.

The qualitative assessment demonstrated that the proposed model could produce high-quality images, superior to the low-resolution (LR) counterparts. A preliminary PSNR evaluation indicated that our model underperformed compared to the original EDVR model, but this could be attributed to resource and time constraints. Implementation was not seamless; both models have significant computational requirements, and our available resources were limited. Moreover, the complexity of EDVR's documentation presented initial hurdles.

The study was constrained by limited computational resources and time. Moreover, we focused primarily on qualitative measures, leaving scope for future quantitative assessments. The model could also be tested on a wider variety of datasets and conditions.

The fusion of YOLOv8-seg and EDVR presents a novel approach to SR, merging the capabilities of object detection and SR uniquely. While the results are preliminary, they offer a compelling foundation for future research in computer vision.

In summary, the proposed model demonstrates the potential to advance the state of the art in super-resolution methodologies by integrating feature mask information. However, further research is essential to fine-tune the model and evaluate its performance under different conditions and metrics.

X. Acknowledgment

I would like to thank my supervisor Dr Yuhang Ye for providing invaluable support and help in developing and writing this research work. I am grateful to my supervisor for generously giving me directions and suggestions for my paper revisions and the direction of the work and paper submissions. I must also offer my thanks and appreciation to Dr Enda Fallon for all of his support and help.

References

[1] https://docs.ultralytics.com/yolov5/tutorials/architecture_description / (Accessed: 15 July 2023).

[2] Ultralytics (2023) *Yolov8, Ultralytics YOLOv8 Docs*. Available at: https://docs.ultralytics.com/models/yolov8/#usage (Accessed: 12 June 2023).

[3] Ayas, S., & Ekinci, M. (2020). Single image super resolution using dictionary learning and sparse coding with multi-scale and multi-directional Gabor feature representation. Information Sciences, 512, 1264–1278.

[4] Dai, J. *et al.* (2017) 'Deformable Convolutional Networks', *2017 IEEE International Conference on Computer Vision (ICCV)* [Preprint]. doi:10.1109/iccv.2017.89.

[5] Bashir, S. M. A., Wang, Y., Khan, M., & Niu, Y. (2021). A comprehensive review of deep learning-based single image super-resolution. PeerJ Computer Science, 7, e621.

[6] Lim, B. et al. (2017) Enhanced Deep Residual Networks for Single Image Super-Resolution, arXiv.org. Available at: https://arxiv.org/abs/1707.02921v1.

[7] Ledig, C. et al. (2016) Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, arXiv.org. Available at: https://arxiv.org/abs/1609.04802v5.

[8] Wang, X. *et al.* (2019) 'EDVR: Video restoration with enhanced deformable convolutional networks', *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* [Preprint]. doi:10.1109/cvprw.2019.00247.

[9] Dong, C., Loy, C. C., & Tang, X. (2016). Accelerating the super-resolution convolutional neural network. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14, 391–407.

[10] Geng, Z., Liang, L., Ding, T., & Zharkov, I. (2022). Rstt: Real-time spatial temporal transformer for space-time video super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 17441-17451).

[11] Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. Machines, 11(7), 677.

[12] Kaur, H., Koundal, D., & Kadyan, V. (2021). Image fusion techniques: A survey. Archives of Computational Methods in Engineering, 28, 4425–4447.

[13] Kim, S., Jun, D., Kim, B.-G., Lee, H., & Rhee, E. (2021). Single image super-resolution method using cnn-based lightweight neural networks. Applied Sciences, 11(3), 1092.

[14] Kim, J., Jung K. Lee, and Kyoung Mu Lee (2016). Deeply-Recursive Convolutional Network for Image Super-Resolution.

[15] White, E.P. *et al.* (2010) “Integrating spatial and temporal approaches to understanding species richness,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1558), pp. 3633–3643. doi:10.1098/rstb.2010.0280.

[16] Le, H. T., Phung, S. L., & Bouzerdoum, A. (2021). A fast and compact deep Gabor network for micro-Doppler signal processing and human motion classification. IEEE Sensors Journal, 21(20), 23085–23097.

[17] Liu, H., Ruan, Z., Zhao, P., Dong, C., Shang, F., Liu, Y., Yang, L., & Timofte, R. (2022). Video super-resolution based on deep learning: A comprehensive survey. Artificial Intelligence Review, 55(8), 5981–6035.

[18] Liu, X., Shi, K., Wang, Z., & Chen, J. (2021). Exploit camera raw data for video super-resolution via hidden Markov model inference. IEEE Transactions on Image Processing, 30, 2127–2140.

[19] Singh, A., & Singh, J. (2019). Review and comparative analysis of various image interpolation techniques. 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 1, 1214–1218.

[20] Huang, Y.-W. and Chen, J. (2022) *Improved EDVR model for robust and efficient video super-resolution*, *ieeexplore.ieee.org*. Available at: https://ieeexplore.ieee.org/document/9707569/ (Accessed: 10 July 2023).

[21] Lu, Z. *et al.* (2023) *Can sam boost video super-resolution?*, *arXiv.org*. Available at: https://arxiv.org/abs/2305.06524v2 (Accessed: 05 July 2023).

[22] Kirillov, A.M. *et al.* (2023b) “Segment anything,” *arXiv (Cornell University)* [Preprint]. Available at: https://doi.org/10.48550/arxiv.2304.02643.

[23] Vlaović, J., Žagar, D., Rimac-Drlje, S., & Vranješ, M. (2021). Evaluation of objective video quality assessment methods on video sequences with different spatial and temporal activity encoded at different spatial resolutions. International Journal of Electrical and Computer Engineering Systems, 12(1), 1–9.

[24] Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. In Proceedings of the IEEE International Conference on Computer Vision (pp. 3730-3738). Wang, X., Chang, K., *et al.* (2019) *EDVR: Video Restoration with Enhanced Deformable Convolutional Networks*. Available at: https://xinntao.github.io/projects/EDVR (Accessed: June 2023).

[25] Xinntao (no date) *Xinntao/EDVR: EDVR has been merged into BASICSR.*, *GitHub*. Available at: https://github.com/xinntao/EDVR (Accessed: 10 July 2023).

[26] Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., Tang, M., & Wang, J. (2023). Fast Segment Anything. ArXiv Preprint ArXiv:2306.12156.

[27] Kim, J., Lee, J.K. and Lee, K.M. (2015) “Accurate Image Super-Resolution Using Very Deep Convolutional Networks,” *arXiv (Cornell University)* [Preprint]. Available at: https://doi.org/10.48550/arxiv.1511.04587.

[28] Kang, J. *et al.* (2020) “Deep Space-Time video upsampling networks,” in *Springer eBooks*, pp. 701–717. Available at: https://doi.org/10.1007/978-3-030-58607-2_41.

[29] Matsui, Y., Hamamoto, T., & Kato, K. (2017). Sketch-based manga retrieval using manga109 dataset with annotation of mangaka styles. In Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval (pp. 399-402).