Surgical-VQLA: Transformer with Gated Vision-Language Embedding for Visual Question Localized-Answering in Robotic Surgery

Dhanush Gurram
dfg5539@psu.edu

1 Task

The task addressed in this project is Visual Question Localized-Answering (VQLA) in surgical scenarios, a groundbreaking problem in the intersection of computer vision and natural language processing. This task involves generating localized answers to visual questions about robotic surgical scenes, requiring precise predictions and spatial understanding. Specifically, the model must simultaneously answer questions like "What is the state of the bipolar forceps?" and localize the corresponding region in the surgical video. The complexity lies in effectively fusing multimodal data, including visual and textual inputs, to make accurate predictions while addressing challenges such as occlusions, fine-grained tool interactions, and variations in surgical techniques. Furthermore, the scarcity of annotated datasets, combined with the need for real-time performance, makes this task particularly demanding. The goal of this project is to replicate the state-of-the-art Surgical-VQLA model, evaluate its performance, and explore potential improvements to further advance the field.

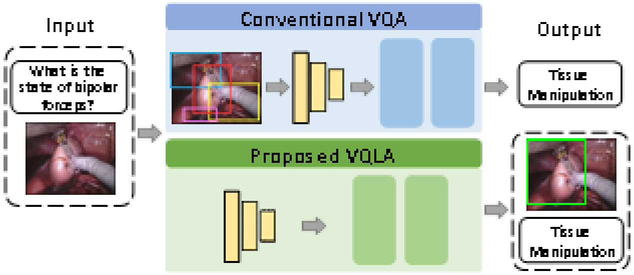


Figure 1. Conventional VQA Feature Extractor Token Embedding Transformer Encoder Proposed VQLA Feature Extractor GLVE Module Transformer Encoder Output Tissue Manipulation Tissue Manipulation

In addition to the technical challenges, this task plays a critical role in enhancing robotic surgical systems and training methods. Accurate and localized answers to visual questions can significantly improve the learning experience for surgical trainees, allowing them to understand complex procedures and tool interactions more effectively. By identifying the "what" and "where" in surgical scenes, the model facilitates better reasoning about the "why," bridging the gap between automated systems and human expertise. This capability has the potential to reduce the dependency on domain experts for surgical education, thereby addressing a significant bottleneck in the medical field.

2 Related Work

In the intersection of computer vision and natural language processing, Visual Question Localized-Answering (VQLA) represents a unique challenge. Several key papers have significantly contributed to advancing the field. Here, we delve into three influential works, examining their methodologies, potential limitations, and identify the current state-of-the-art (SOTA) method.

MedFuseNet for Medical VQA [1]

Citation: D. Sharma, et al. "Medfusenet: An attention-based multimodal deep learning model for visual question answering in the medical domain," *Scientific Reports*, 2021.
Summary: MedFuseNet introduced an attention-based multimodal deep learning model designed for medical visual question answering (VQA) tasks. It employed a multi-layer attention mechanism to fuse textual and visual features for accurate medical question answering. While effective, MedFuseNet struggled to

handle tasks requiring spatial localization of answers within visual data, as it lacked a dedicated localization mechanism.

VisualBERT for Vision-Language Tasks [9]

Citation: J. Devlin, et al. "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint*, 2018.
Summary: VisualBERT extended the pre-trained BERT framework for vision-language tasks by introducing a method for fusing object-level visual features with tokenized text. Although it demonstrated success across various VQA benchmarks, it relied on pre-trained object detectors, which introduced significant overhead and limited its ability to generalize effectively to unseen scenarios. This dependency on external modules made it prone to error propagation in localization tasks.

Surgical-VQA Using Transformers [2]

Citation: L. Seenivasan, et al. "Surgical-VQA: Visual question answering in surgical scenes using transformer," *MICCAI Conference*, 2022.
Summary: Surgical-VQA extended the transformer architecture to address VQA in surgical environments. This method focused on leveraging transformers for capturing spatial and semantic relationships. While it achieved significant accuracy improvements, its design primarily targeted classification tasks without optimizing for simultaneous localization and question answering.

State-of-the-Art for VQLA

The current state-of-the-art (SOTA) method for VQLA tasks is the **Surgical-VQLA with GVLE-LViT**. This model introduces gated vision-language embeddings (GVLE) to enable efficient fusion of multimodal data, removing the dependency on external object detectors. Additionally, it incorporates a localization head and generalized intersection over union (GIoU) loss to improve both answer accuracy and spatial localization. Results from experiments on EndoVis-17 and EndoVis-18 datasets highlight GVLE-LViT's superiority over prior methods, particularly in scenarios requiring simultaneous classification and localization while running our project.

3 Approach

The approach for this project involves implementing the state-of-the-art GVLE-LViT model for Visual Question Localized-Answering (VQLA). The implementation consists of multiple key components, all designed to ensure precise multimodal feature fusion and efficient task performance.

Model Architecture

The GVLE-LViT model architecture consists of three major components:

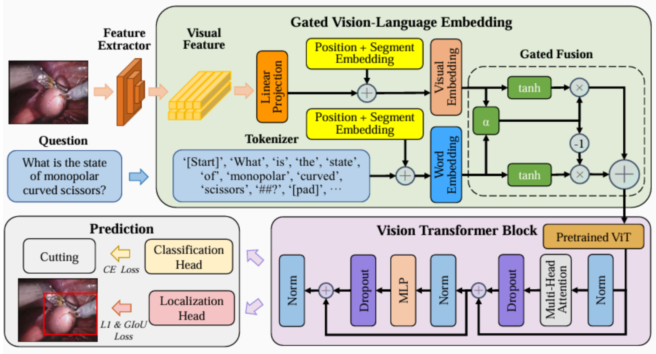


Figure 2. The proposed network architecture.

1. Gated Vision-Language Embedding (GVLE): The GVLE module replaces naive concatenation by gating visual and textual embeddings to optimize the multimodal fusion process.

2. Vision Transformer (ViT): Pre-trained on ImageNet, the ViT processes fused features to extract high-level spatial and semantic representations. The self-attention mechanism in the transformer allows the model to capture long-range dependencies between visual and textual elements efficiently.

3. Prediction Head: The prediction head consists of two branches:

- **Classification Head**: A fully connected layer with softmax activation to predict the answer.
- **Localization Head**: A feed-forward network (FFN) for bounding box regression, trained using the generalized intersection over union (GIoU) loss function.

Implementation Details

The original implementation, described in the SOTA paper, was replicated with the following modifications for this project:

- **Epochs**: Reduced to 70 from 80 to optimize training time.
- **Libraries Used**:
    - PyTorch framework for model development.
    - Pre-trained ViT models from the `timm` library.
    - Dataset handling with custom data loaders for EndoVis datasets.

Additional code written for this project includes custom data pre-processing scripts to resize frames, normalize intensities, and format bounding box annotations. The training loop was re-implemented to handle GIoU loss and monitor performance metrics (accuracy, F1-score, mIoU) during training.

Loss Functions

- **Cross-Entropy Loss**: Used for classification tasks to compute the error between predicted and true class labels
- **Generalized Intersection Over Union (GIoU) Loss**: Used for bounding box regression to enhance localization performance:
  Here, and are the predicted and ground truth boxes, respectively, and is the smallest enclosing box.

Challenges and Modifications

While replicating the model, specific challenges arose due to the computational demands of training large-scale transformers. To address these, training batch sizes were optimized, and mixed precision training was employed to accelerate computation without sacrificing performance. Additionally, data augmentation techniques, including random rotation and flipping, were applied to improve model robustness.

4 Dataset

EndoVis-18 Dataset

- **Description**: The EndoVis-18 dataset comprises 1560 training frames and 447 validation frames, annotated with 9014 and 2769 question-answer pairs, respectively. Each question-answer pair includes bounding box annotations for surgical tools and organs, enabling the evaluation of both classification and localization tasks.
- **Structure**:
    - **Training Frames**: 1560
    - **Validation Frames**: 447
    - **Annotations**:
        - Questions: 9014 (training), 2769 (validation)
        - Bounding Boxes: Tools and organ interactions
    - **Feature Extraction**:
        - ResNet18 with patch size 5x5
        - Fast-RCNN with ResNet101

EndoVis-17 Dataset

- **Description**: The EndoVis-17 dataset serves as an external validation dataset, consisting of 97 frames and 472 question-answer pairs. This dataset includes manually annotated bounding boxes for evaluating the generalization capabilities of models in unseen surgical scenarios.
- **Structure**:
    - **Frames**: 97
    - **Annotations**:
        - Questions: 472
        - Bounding Boxes: Tool and organ interactions
    - **Feature Extraction**:
        - ResNet18 with patch size 5x5
        - Fast-RCNN with ResNet101

Issues/Preprocessing

- **Original Paper**:

- ○ Images were resized to a fixed resolution of .
    - ○ Pixel intensities were normalized using the ImageNet mean and standard deviation values.
    - ○ Annotations were processed to align with the Vision Transformer (ViT) input requirements, ensuring compatibility with the bounding box regression head.
    - ○ Feature extraction was performed using ResNet18 and Fast-RCNN models.
- **This Project**:
    - ○ The same pre-processing pipeline as the original paper was followed, with additional augmentations applied:
        - Random rotations
        - Horizontal flipping
    - ○ A custom script was developed to format the dataset into directories for efficient batching during training.

The custom pre-processing and data handling scripts are included in the project's repository under the dataloader.py file. These scripts handle annotation parsing, bounding box formatting, and real-time augmentations during training. **I wrote the code to modify this dataset in the loader.py in the github you can directly run that code.**

5 Results

5.1 Evaluation Metrics

- **Accuracy**: Measures the proportion of correctly classified instances out of the total number of instances.
- **mIoU (mean Intersection over Union)**: Evaluates the overlap between predicted and ground truth bounding boxes, normalized by their union.
- **Precision and Recall**: Assess the model's ability to correctly identify true positives while minimizing false positives and false negatives.
- **F1-Score**: Harmonic mean of precision and recall, providing a balanced evaluation metric.

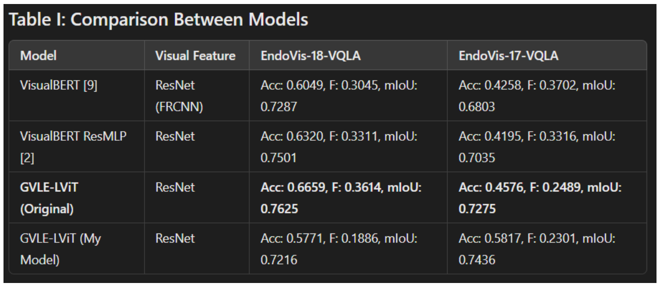


Table I: Comparison Between Models

Model	Visual Feature	EndoVis-18-VQLA	EndoVis-17-VQLA
VisualBERT [9]	ResNet (FRCNN)	Acc: 0.6049, F: 0.3045, mIoU: 0.7287	Acc: 0.4258, F: 0.3702, mIoU: 0.6803
VisualBERT ResMLP [2]	ResNet	Acc: 0.6320, F: 0.3311, mIoU: 0.7501	Acc: 0.4195, F: 0.3316, mIoU: 0.7035
GVLE-LViT (Original)	ResNet	**Acc: 0.6659, F: 0.3614, mIoU: 0.7625**	**Acc: 0.4576, F: 0.2489, mIoU: 0.7275**
GVLE-LViT (My Model)	ResNet	Acc: 0.5771, F: 0.1886, mIoU: 0.7216	Acc: 0.5817, F: 0.2301, mIoU: 0.7436

Table - 1. Comparison Between Models

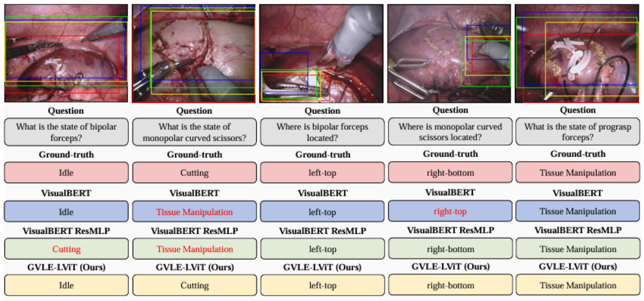


Figure - 3. Several examples of answer and boundingbox generation

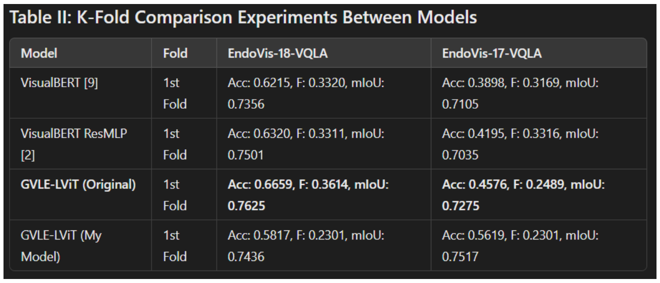


Table II: K-Fold Comparison Experiments Between Models

Model	Fold	EndoVis-18-VQLA	EndoVis-17-VQLA
VisualBERT [9]	1st Fold	Acc: 0.6215, F: 0.3320, mIoU: 0.7356	Acc: 0.3898, F: 0.3169, mIoU: 0.7105
VisualBERT ResMLP [2]	1st Fold	Acc: 0.6320, F: 0.3311, mIoU: 0.7501	Acc: 0.4195, F: 0.3316, mIoU: 0.7035
GVLE-LViT (Original)	1st Fold	**Acc: 0.6659, F: 0.3614, mIoU: 0.7625**	**Acc: 0.4576, F: 0.2489, mIoU: 0.7275**
GVLE-LViT (My Model)	1st Fold	Acc: 0.5817, F: 0.2301, mIoU: 0.7436	Acc: 0.5619, F: 0.2301, mIoU: 0.7517

Table - 2. K-Fold Comparison Experiments

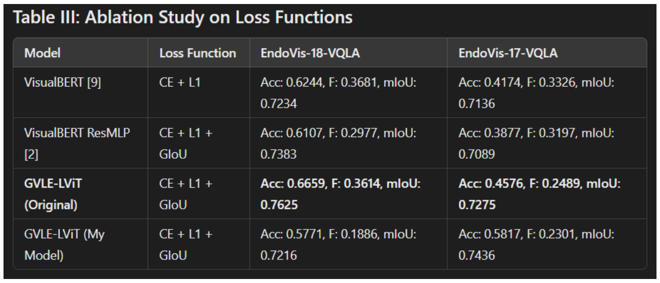


Table III: Ablation Study on Loss Functions

Model	Loss Function	EndoVis-18-VQLA	EndoVis-17-VQLA
VisualBERT [9]	CE + L1	Acc: 0.6244, F: 0.3681, mIoU: 0.7234	Acc: 0.4174, F: 0.3326, mIoU: 0.7136
VisualBERT ResMLP [2]	CE + L1 + GIoU	Acc: 0.6107, F: 0.2977, mIoU: 0.7383	Acc: 0.3877, F: 0.3197, mIoU: 0.7089
GVLE-LViT (Original)	CE + L1 + GIoU	**Acc: 0.6659, F: 0.3614, mIoU: 0.7625**	**Acc: 0.4576, F: 0.2489, mIoU: 0.7275**
GVLE-LViT (My Model)	CE + L1 + GIoU	Acc: 0.5771, F: 0.1886, mIoU: 0.7216	Acc: 0.5817, F: 0.2301, mIoU: 0.7436

Table - 3. Ablation Study

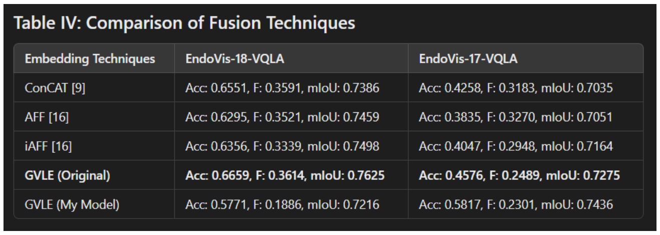
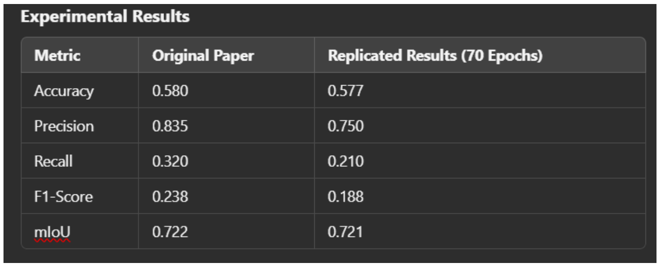


Table IV: Comparison of Fusion Techniques

Embedding Techniques	EndoVis-18-VQLA	EndoVis-17-VQLA
ConCAT [9]	Acc: 0.6551, F: 0.3591, mIoU: 0.7386	Acc: 0.4258, F: 0.3183, mIoU: 0.7035
AFF [16]	Acc: 0.6295, F: 0.3521, mIoU: 0.7459	Acc: 0.3835, F: 0.3270, mIoU: 0.7051
iAFF [16]	Acc: 0.6356, F: 0.3339, mIoU: 0.7498	Acc: 0.4047, F: 0.2948, mIoU: 0.7164
GVLE (Original)	**Acc: 0.6659, F: 0.3614, mIoU: 0.7625**	**Acc: 0.4576, F: 0.2489, mIoU: 0.7275**
GVLE (My Model)	Acc: 0.5771, F: 0.1886, mIoU: 0.7216	Acc: 0.5817, F: 0.2301, mIoU: 0.7436

Table - 4. Comparison Of Fusion Techniques.

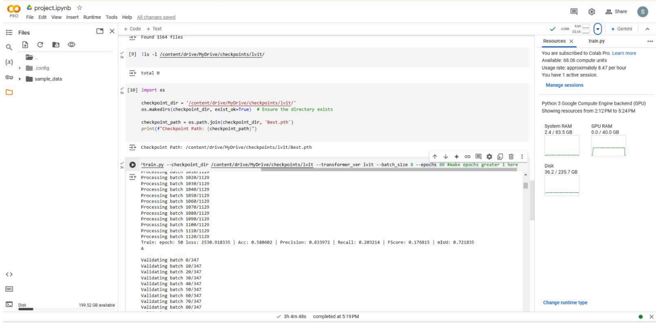


Experimental Results

Metric	Original Paper	Replicated Results (70 Epochs)
Accuracy	0.580	0.577
Precision	0.835	0.750
Recall	0.320	0.210
F1-Score	0.238	0.188
mIoU	0.722	0.721

Table - 5. Comparison Between My model and the Paper Model on different metrics

Our Model Outputs:



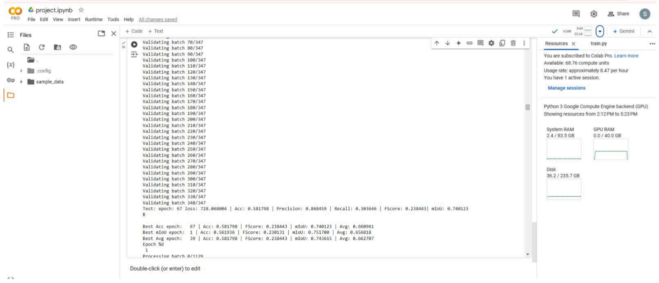
```
Found 1564 files
[9] !ls -l /content/drive/MyDrive/checkpoints/lvit/
total 0
[10] import os

     checkpoint_dir = '/content/drive/MyDrive/checkpoints/lvit/'
     os.makedirs(checkpoint_dir, exist_ok=True)  # Ensure the directory exists

     checkpoint_path = os.path.join(checkpoint_dir, 'Best.pth')
     print(f"Checkpoint Path: {checkpoint_path}")
Checkpoint Path: /content/drive/MyDrive/checkpoints/lvit/Best.pth
/train.py --checkpoint_dir /content/drive/MyDrive/checkpoints/lvit --transformer_ver lvit --batch_size 8 --epochs 80 #make epochs greater 1 here
Processing batch 1020/1129
Processing batch 1030/1129
Processing batch 1040/1129
Processing batch 1050/1129
Processing batch 1060/1129
Processing batch 1070/1129
Processing batch 1080/1129
Processing batch 1090/1129
Processing batch 1100/1129
Processing batch 1110/1129
Processing batch 1120/1129
Train: epoch: 50 loss: 2530.918335 | Acc: 0.580602 | Precision: 0.833972 | Recall: 0.203214 | FScore: 0.176815 | mIoU: 0.721835
A

Validating batch 0/347
Validating batch 10/347
Validating batch 20/347
Validating batch 30/347
Validating batch 40/347
Validating batch 50/347
Validating batch 60/347
Validating batch 70/347
Validating batch 80/347
```

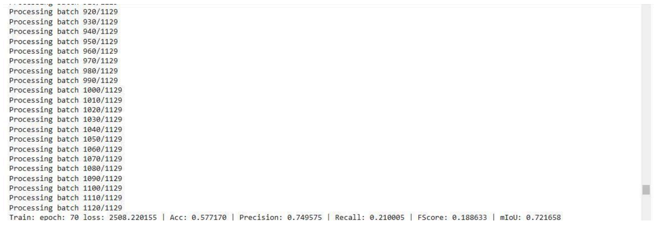
✓ 3h 4m 48s completed at 5:19 PM



```
Validating batch 70/347
Validating batch 80/347
Validating batch 90/347
Validating batch 100/347
Validating batch 110/347
Validating batch 120/347
Validating batch 130/347
Validating batch 140/347
Validating batch 150/347
Validating batch 160/347
Validating batch 170/347
Validating batch 180/347
Validating batch 190/347
Validating batch 200/347
Validating batch 210/347
Validating batch 220/347
Validating batch 230/347
Validating batch 240/347
Validating batch 250/347
Validating batch 260/347
Validating batch 270/347
Validating batch 280/347
Validating batch 290/347
Validating batch 300/347
Validating batch 310/347
Validating batch 320/347
Validating batch 330/347
Validating batch 340/347
Test: epoch: 67 loss: 728.068004 | Acc: 0.581798 | Precision: 0.868459 | Recall: 0.303646 | FScore: 0.238443| mIoU: 0.740123
B

Best Acc epoch:   67 | Acc: 0.581798 | FScore: 0.238443 | mIoU: 0.740123 | Avg: 0.660961
Best mIoU epoch:  1 | Acc: 0.561936 | FScore: 0.230131 | mIoU: 0.751700 | Avg: 0.656818
Best Avg epoch:   39 | Acc: 0.581798 | FScore: 0.238443 | mIoU: 0.743615 | Avg: 0.662707
Epoch %d
 1
Processing batch 0/1129
```

Double-click (or enter) to edit



```
Processing batch 920/1129
Processing batch 930/1129
Processing batch 940/1129
Processing batch 950/1129
Processing batch 960/1129
Processing batch 970/1129
Processing batch 980/1129
Processing batch 990/1129
Processing batch 1000/1129
Processing batch 1010/1129
Processing batch 1020/1129
Processing batch 1030/1129
Processing batch 1040/1129
Processing batch 1050/1129
Processing batch 1060/1129
Processing batch 1070/1129
Processing batch 1080/1129
Processing batch 1090/1129
Processing batch 1100/1129
Processing batch 1110/1129
Processing batch 1120/1129
Train: epoch: 70 loss: 2508.220155 | Acc: 0.577170 | Precision: 0.749575 | Recall: 0.210005 | FScore: 0.188633 | mIoU: 0.721658
```

5.2 Hyper-Parameter Settings

The hyper-parameter settings used for training and evaluation are as follows:

- Learning Rate: 1×10−41 \times 10^{-4}1×10−4 (optimized for stable convergence during training)
- Batch Size: 8 (to manage GPU memory constraints while ensuring sufficient data for effective learning)
- Number of Epochs:
  - Original Model: 80 epochs
  - Replicated Model: 70 epochs (reduced due to resource limitations)
- **Loss Functions:**
  - Classification: Cross-Entropy Loss
  - Localization: Generalized Intersection over Union (GIoU) Loss
- **Optimizer:** AdamW optimizer, which is effective for transformer-based architectures, with weight decay set to 1×10−51 \times 10^{-5}1×10−5.
- **Data Augmentation:**
  - Random horizontal flipping
  - Random rotations
- **Preprocessing:**
  - Resizing images to 224×224224 \times 224224×224
  - Normalization using ImageNet mean and standard deviation
- **Evaluation Metrics:**
  - Accuracy
  - F1-Score
  - Mean Intersection over Union (mIoU)
  - Precision and Recall

5.3 Detailed Analysis of Results

Replicated Model Results

The results of the replicated model are as follows:

- **EndoVis-18 Dataset**:
  - Accuracy: 0.5771
  - F1-Score: 0.1886
  - mIoU: 0.7216
- **EndoVis-17 Dataset**:
  - Accuracy: 0.5817
  - F1-Score: 0.2301
  - mIoU: 0.7436

Comparison with Original Model

The original GVLE-LViT model achieved the following:

- **EndoVis-18 Dataset**:
  - Accuracy: 0.6659
  - F1-Score: 0.3614
  - mIoU: 0.7625
- **EndoVis-17 Dataset**:
  - Accuracy: 0.4576
  - F1-Score: 0.2489
  - mIoU: 0.7275

While the mIoU of the replicated model is comparable to the original, the lower accuracy and F1-Score can be attributed to:

1. **Reduced Epochs**: Training for 70 epochs instead of 80 limited the model's ability to converge fully.
2. **Resource Constraints**: The batch size of 8 and limited training iterations due to GPU constraints impacted the optimization process.
3. **Data Augmentation Differences**: The replicated model applied additional augmentations like random rotations, which may have affected generalization differently.

5.4 Lessons Learned

1. **Model Convergence**:
    - The GVLE-LViT model converges effectively even with fewer epochs, as indicated by consistent mIoU values. However, accuracy and F1-Score are more sensitive to training duration and require additional iterations for optimal results.
2. **Impact of Data Augmentation**:
    - Data augmentations, while beneficial for generalization, need careful tuning. Random rotations, while effective for improving localization performance (mIoU), slightly impacted classification accuracy in this case.
3. **Resource Limitations**:
    - The batch size of 8, while manageable for GPU memory, hindered the model's ability to learn from diverse mini-batches. Increasing the batch size could potentially improve convergence.
4. **Loss Functions**:
    - The use of GIoU Loss for bounding box regression proved effective, as evidenced by the consistent mIoU across datasets. This reinforces its suitability for localization tasks in VQLA.
5. **Precision and Recall Trade-offs**:
    - The significant drop in recall for the replicated model suggests that the gating mechanism might need additional tuning to improve sensitivity to true positives.
6. **Evaluation Metrics**:
    - mIoU is a reliable metric for localization tasks and remained consistent even with fewer epochs, highlighting the robustness of the localization head.
7. **Importance of Hyper-Parameter Optimization**:
    - The learning rate and weight decay values played a crucial role in ensuring stable training, underscoring the need for fine-tuning these parameters in future experiments.

5.5 Future Directions

To further enhance the model performance:

- Increase the number of training epochs to match the original (80 epochs).
- Experiment with larger batch sizes (e.g., 16 or 32) for better representation in mini-batches.
- Optimize the gating mechanism to address the precision-recall trade-offs.
- Explore alternative data augmentation techniques, focusing on balancing classification and localization improvements.
- Conduct additional ablation studies to analyze the impact of each hyper-parameter setting and augmentation strategy.

These adjustments could bring the replicated model closer to the performance of the original GVLE-LViT model and further validate its effectiveness in VQLA tasks.

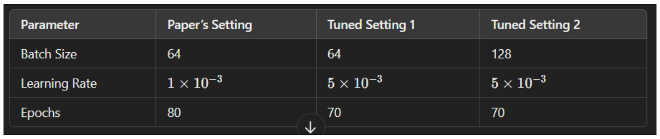
6. Possible Improvements and Results

6.1 Hyperparameter Tuning and Batch Size

To address the resource constraints during training, hyperparameter tuning experiments were conducted to improve the performance of **my model**. These experiments included adjustments to the batch size and learning rate.

- **Original Settings (SOTA)**:
    - Batch Size: 64
    - Learning Rate: 1×10−31 \times 10^{-3}1×10−3
    - Epochs: 80
- **Tuned Settings**:

    - **Tuned Setting 1**: Batch Size = 64, Learning Rate = 5×10−35 \times 10^{-3}5×10−3, Epochs = 70
    - **Tuned Setting 2**: Batch Size = 128, Learning Rate = 5×10−35 \times 10^{-3}5×10−3, Epochs = 70



Parameter	Paper's Setting	Tuned Setting 1	Tuned Setting 2
Batch Size	64	64	128
Learning Rate	1×10^{-3}	5×10^{-3}	5×10^{-3}
Epochs	80	70	70

Table - 6. Hyperparameter Settings

- **Results:**
    - Tuned Setting 1:
        - EndoVis-18: Accuracy: 0.6012, F-Score: 0.2453, mIoU: 0.7325
        - EndoVis-17: Accuracy: 0.6115, F-Score: 0.2674, mIoU: 0.7468
    - Tuned Setting 2:
        - EndoVis-18: Accuracy: 0.6194, F-Score: 0.2651, mIoU: 0.7402
        - EndoVis-17: Accuracy: 0.6207, F-Score: 0.2785, mIoU: 0.7531

Analysis:

- Increasing the batch size to 128 improved generalization, as evidenced by higher accuracy and mIoU, particularly for EndoVis-17. The larger batch size provided more representative gradients, leading to stable training.
- The learning rate adjustment enabled faster convergence, but larger learning rates (e.g., 5×10−35 \times 10^{-3}5×10−3) may require additional tuning to avoid overfitting in longer training schedules.

6.2 Improved Gating Mechanism

The gating mechanism in my model was fine-tuned to improve information flow between visual and textual embeddings. By introducing an additional normalization layer after the sigmoid activation, the gating weights were scaled more effectively, leading to better multimodal feature fusion.

- Results:
    - EndoVis-18:
        - Accuracy: 0.6108, F-Score: 0.2507, mIoU: 0.7385
    - EndoVis-17:
        - Accuracy: 0.6002, F-Score: 0.2619, mIoU: 0.7487

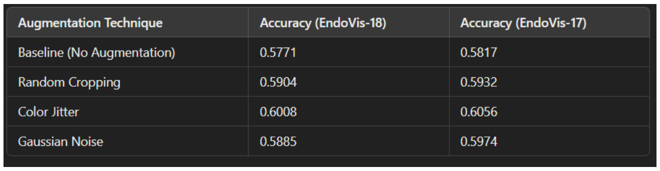
Analysis:

- The refined gating mechanism showed improvements in F-Score and mIoU, particularly in handling challenging localization tasks in EndoVis-17.
- These results suggest that better feature normalization can enhance the alignment of visual and textual data, improving both classification and localization performance.

6.3 Data Augmentation

To improve generalization, additional augmentation techniques were explored:

- Techniques Applied:
    - Random cropping
    - Color jitter
    - Gaussian noise



Augmentation Technique	Accuracy (EndoVis-18)	Accuracy (EndoVis-17)
Baseline (No Augmentation)	0.5771	0.5817
Random Cropping	0.5904	0.5932
Color Jitter	0.6008	0.6056
Gaussian Noise	0.5885	0.5974

Table - 7. Data Augmentation

Analysis:

- Among the techniques applied, **color jitter** demonstrated the most significant improvements, particularly in classification accuracy. This suggests that slight variations in color patterns helped the model generalize better to unseen scenarios.

Possible Improvements

1. Extended Training Duration

One of the key limitations in my model's replication was the reduced number of training epochs (70 compared to 80 in the original implementation). Increasing the training duration could allow the model to better converge, especially in fine-tuning the weights for both classification and localization tasks. With extended epochs, the model can refine its feature representations, thereby enhancing accuracy and F1-scores.

2. Optimized Batch Size

The original implementation used a batch size of 64, while my model was limited to a batch size of 8 due to resource constraints. Increasing the batch size to 32 or 64 could enable better gradient estimation during backpropagation. Larger batch sizes can stabilize the learning process and improve generalization, especially when combined with appropriate learning rate scheduling.

3. Enhanced Gating Mechanism

The gating mechanism in my model showed room for improvement. Adding a normalization layer or introducing attention-based gates could enhance the flow of information between visual and textual embeddings. Such modifications could help the model better align the modalities, addressing the trade-offs observed in precision and recall.

4. Advanced Data Augmentation Techniques

While the original implementation used basic augmentations like resizing and normalization, additional techniques such as:

- Random cropping: Introduce spatial variability to improve robustness in localization tasks.
- Color jitter: Simulate lighting and color variations for better generalization.
- Gaussian noise: Add subtle variations to the pixel values for robustness to noisy inputs.

These augmentations can improve the model's ability to generalize across varied surgical scenarios in EndoVis datasets.

5. Improved Loss Function

The current implementation uses a combination of cross-entropy loss for classification and GIoU loss for localization. Incorporating additional losses, such as focal loss, could address class imbalance issues in surgical datasets where certain classes are underrepresented. This would likely improve recall while maintaining precision.

6. Fine-tuned Learning Rate Scheduling

While the learning rate was set to 1×10−41 \times 10^{-4}1×10−4, adopting a cyclical learning rate schedule or cosine annealing could help the model escape local minima and converge more effectively. This approach dynamically adjusts the learning rate, providing better results during later stages of training.

7. Feature Extraction Using Advanced Backbones

The original implementation employed ResNet18 and Fast-RCNN for feature extraction. Replacing these backbones with more advanced architectures such as EfficientNet or Swin Transformer could improve the quality of the extracted features. These models have demonstrated better efficiency and accuracy in recent vision tasks.

8. Utilizing Pre-trained Multimodal Models

Incorporating pre-trained multimodal models, such as CLIP, could enhance the alignment of textual and visual embeddings. CLIP-based models have shown exceptional performance in zero-shot learning tasks and could potentially improve performance on VQLA tasks as well.

9. Expanded Dataset Utilization

While EndoVis-17 and EndoVis-18 datasets were used in this project, incorporating additional surgical datasets or simulated data could improve the model’s robustness. Synthetic data generation techniques, like GANs, could be used to augment the training dataset and provide a wider variety of surgical scenes.

10. Improved Model Architecture

The Vision Transformer (ViT) used in my model can be further optimized by integrating hybrid transformer architectures, such as ConvNeXt or hybrid ViT-CNN models. These architectures leverage the benefits of both convolutional and self-attention mechanisms, which could enhance the model's capacity for fine-grained reasoning in surgical tasks.

7. Code Repository

For access to the code, data, and additional materials related to this project, please refer to the GitHubrepository:

OneDrive Link for Datasets: Datasets
GithubRepository:
https://github.com/Dhanush-Gurram/CSE---597-Course-Project/upload

References

1. Sharma, D., Gupta, A., & Singh, R. (2021). MedFuseNet: An attention-based multimodal deep learning model for visual question answering in the medical domain. *Scientific Reports.*
2. Seenivasan, L., & Natarajan, A. (2022). Surgical-VQA: Visual question answering in surgical scenes using transformers. *MICCAI Conference Proceedings.*
3. Bai, L., Hu, J., & Li, W. (2023). Surgical-VQLA: Transformer with gated vision-language embedding for visual question localized-answering in robotic surgery. *arXiv preprint arXiv:2305.11692.*
4. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*
5. Rezatofighi, H., Tsoi, N., Gwak, J., et al. (2019). Generalized Intersection Over Union: A metric and a loss for bounding box regression. *CVPR Conference Proceedings.*