EEE6778 - RetinAI Deliverable 2

Soroush Saririan
UF ADS
Gainesville, FL

***Abstract*—In this paper we will the implementation of the RetinAI project. This is a project that utilizes two Diabetic Retinopathy datasets along with Vision Language Models (VLMs) to create a screening tool. By tuning a LLaVA model on these datasets, the final project will produce classification and explanations for these medical images.**

I. Problem Summary

Diabetic Retinopathy is a leading cause behind preventable blindness. This is a significant challenge for public health, and one of the main factors is the very subtle early onset indicators. This is a major issue in areas where healthcare is limited or ares without proper medical facilities. This can lead to delayed diagnosis, resulting in unrepairable eye damage.

To solve this issue, the goal of this project is to use VLMs, specifically the use of LLaVA, which is specifically tailored for medical image datasets. This method will allow for medical professionals to gain a fast and secondary opinion for their initial classification, providing an additional line of defense against preventable blindness.

II. Datasets

This project utilizes two datasets:

1) APTOS 2019 Blindness Detection
2) IDRiD (Indian Diabetic Retinopathy Dataset)

A. APTOS

This dataset is a large scale dataset of retinal fundus images via a prior Kaggle competition.

Dataset Info:

- About 10 GB of images and masks
- 3,662 training images
- 1,928 testing images
- Diagnosis scale from 0-4
- Images in .png format
- Labels in a .csv file, with corresponding image ID

This dataset will be utilized for training the final classification model. The diversity in the data and the larger size are great to help with generalization on unseen images, which is essential for this project.

This dataset will also be used for benchmarking, and will be further fine tuned to this task via additional training from the IDRiD dataset.

B. IDRiD

This is a smaller dataset that contains more detailed fundus images from a Grand Challenge via IEEE.

Dataset Info:

- 413 training images
- 103 testing images
- Images in .jpg format
- Diagnosis scale from 0-4
- Pixel level segmentation masks for specific retinal lesions

The purpose of this dataset is to cut down on the training time for the final dataset. It will allow for faster and more efficient initial training of the model. This will allow for faster and easier initial prototyping, before working on the larger dataset.

Also due to the pixel level segmentation masks that are in this dataset, it will allow for the VLM to generate feature based explanations in the predictions and summary that are specific to certain lesions such as hemorrhages.

C. Access Methods

The APTOS dataset was downloaded via Kaggle, by registering for a competition. The dataset was downloaded using the Kaggle API or via this link:
https://www.kaggle.com/competitions/aptos2019-blindness-detection.

The IDRiD dataset was downloaded via the IEEE Dataport. This can be directly downloaded via the IEEE webpage after creating an account via this link:
https://ieee-dataport.org/open-access/indian-diabetic-retinopathy-image-dataset-idrid.

D. Preprocessing

The main issue for preprocessing of these two datasets, is the large amount of data with a diagnosis of Grade 0, i.e. no Diabetic Retinopathy. As a result, the classification model can become biased, learning the classification of non Retinopathy very well, but performing poorly in those cases that have very small features that are positive for Diabetic Retinopathy, which are of the utmost clinical importance.

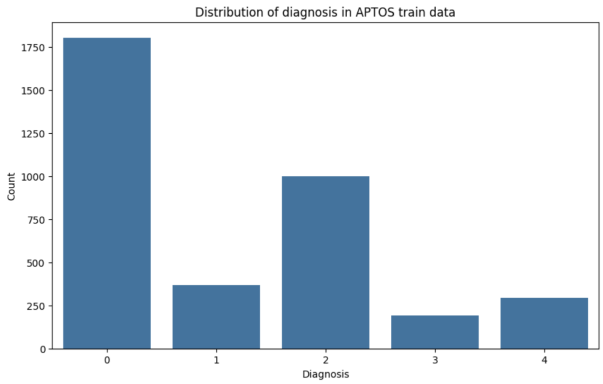


Fig. 1. Distribution of Diagnosis for APTOS

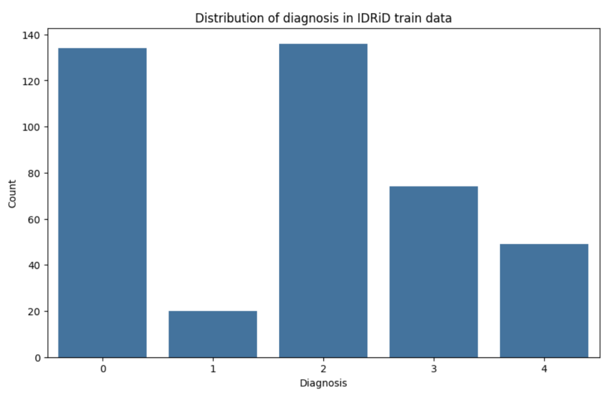


Fig. 2. Distribution of Diagnosis for IDRiD

There are a few methods that can be used to overcome this issue

1) Weighted loss functions to penalize incorrect classifications more harshly
2) SMOTE, syntheticaly creating new images of minority diagnosis grades
3) Data augmentation by creating different versions of preexisting images via rotations or mirroring

Another potential issue would be the variation in the images that are in the datasets.

Certain images are of different quality or lighting, and between the two datasets, the images are not necessarily the same size. This can cause the model to learn features that are not directly applicable to the actual classification, causing an incorrect classification.

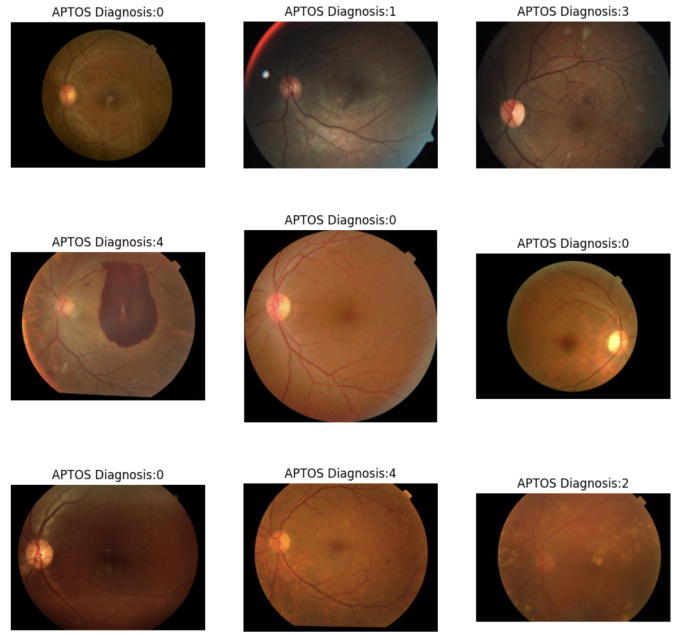


Fig. 3. Image Differences from APTOS Dataset

This can be solved by creating a preprocessing pipeline that does the following:

1) Resizes the images to a uniform dimension
2) Normalize pixel range
3) Potential data augmentation if necessary

E. Current Work

At this point, the current implementation has mainly concerned the training of LLaVA on the IDRID dataset. This has been a challenge mainly due to resources. Implementation was attempted using a M2 Max Macbook Pro, which upon initial research should have been decent for this initial training. However, upon testing system memory became a critical issue.

To solve this use of Google Collab was attempted and that was still having issues. The solution was to use Hipergator and SLURM requests, which proved to be a great solution.

The initial training of the model was done following a project from another course (BME6938), as personal experience with training LLaVA was extremely limited. Using this framework and basic code architecture, a model was created that has average performance. However this is not an issue as this will be used to then train on the APTOS dataset. The purpose of the IDRID dataset training is purely to simplify that much larger task, and to pretrain the LLaVA model on this type of data.

A point to note, is the original proposition of using LLaVA-Med, however this very quickly became something out of the scope of my level of expertise, as just getting access to the model was not possible with my experience and timeframe.

F. Frameworks and Libraries

The implementation of this project centers on fine tuning a pre trained VLM .

The project relies on a conda environment, defined in the environment.yml file, to ensure reproducibility. Key libraries include:

- Core Architecture: PyTorch and the Hugging Face transformers library are the core frameworks for loading, fine-tuning, and running the VLM.
- Performance & Memory: Accelerate and bitsandbytes are used to enable a more effcient 4-bit quantized training on NVIDIA GPUs, drastically reducing memory usage.
- VLM Architecture: The LLaVA library is included to provide the custom model code required to run the llava-hf/llava-1.5-7b-hf architecture.
- Data Handling: pandas is used for managing the CSV label files, while Pillow (PIL) and OpenCV are used for loading and processing images.
- Interface: Gradio is the selecte library for building the final interactive user interface.

Training Setup:

- Hardware: All of the training will be done on Hipergator via the A100 GPU.
- Model: The model to be trained is llava-hf/llava-1.5-7b-hf. This is a powerful VLM that will be fine tuned for the Retinopathy imaging task.
- Two Phase Training Strategy: To prevent overfitting on the small IDRiD dataset, a two phase strategy will be used:
  1) The model will first be tuned on the small, high quality IDRiD dataset (413 images) for only one epoch. The goal of this phase is not to achieve high accuracy, but to teach the model the basic vocabulary and visual features for Diabetic Retinopathy.
  2) The model saved from Phase 1 will then be used as the starting point for a second round of tuning on the larger APTOS dataset for 3-5 epochs, depending on performance. This step will allow the model to generalize its knowledge and will be the primary source of its final accuracy.
- Key Hyperparameters (Planned):
  - model_id: llava-hf/llava-1.5-7b-hf
  - quantization: 4-bit (load_in_4bit=True)
  - precision: bf16=True (optimal for A100 GPUs)
  - per_device_train_batch_size: 1
  - gradient_accumulation_steps: 16 (to simulate an effective batch size of 16)
  - optimizer: adamw_8bit
  - learning_rate: 2e-5
  - gradient_checkpointing: True (to further conserve GPU memory)

Modules for Reproducibility: The project is structured to be fully reproducible by using the following key components:

- environment.yml: Conda environment file that has all required packages and their exact versions, ensuring the software environment can be perfectly replicated.
- src/idrid_setup.py: These scripts convert the raw downloaded datasets and CSV files into the .jsonl format required by the training script.
- src/train_idrid.py and src/test_idrid.py: These scripts contain training and testing processes including all hyperparameters and settings for the tuning process.
- Slurm Folder: The SLURM submission scripts allow for others to determine the resource needs for replication of this project.

G. Ethical Concerns

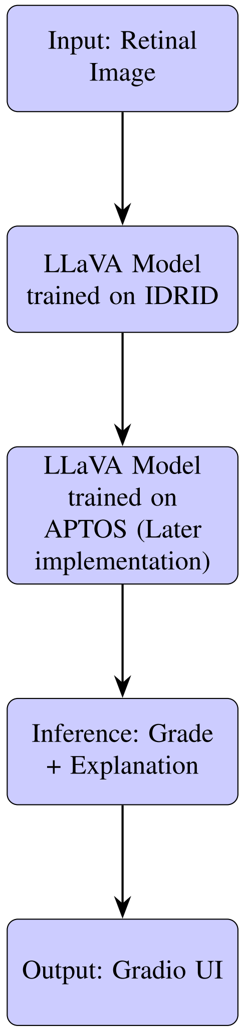
As this project is related to the field of medicine, there is concern about patient confidentiality and ethicality. All data sets for this project were supplied by legitimate competitions, one via Kaggle and the other from IEEE.

All images in these datasets are intended for public research, and any identifiable information is removed (names, clinics, etc).

In terms of ethical bias, one of the most significant challenges is that the IDRiD dataset may not be representative of all ethnic groups because it is a dataset from Indian patients. Therefore, it may not perform as well on certain populations due to factors such as eye color or cameras that were used to take these images.

Finally, the use of this tool is not a replacement for a medical opinion. It is purely a support system or a second opinion. The project may not correctly predict a valid diagnosis, and can cause false positives, leading to patient stress and costs. Or even worse, it could cause false negatives, and if treated as a primary diagnostic tool, can cause missed treatment.

III. Architecture



This project is based around the LLaVA VLM. This hybrid model uses a vision encoder such as CLIP with a LLM allowing for processing of both images and text. The Med portion of LLaVA means that it is trained on medical images and performs better on analysis of these images.

The implementation will use PyTorch for the training and will utilize the Hugging Face library for the model tuning.

The final interface will be created via Gradio. In this interface the user, in this case a healthcare provider, can input a retinal image. The image will then be processed and the model will output a predicted grade for the diagnosis, along with a short description of the reasoning for that diagnosis such as presence of certain lesions.

IV. Gradio Interface

The interface will be created to be as simple and interpretable as possible. The steps for use are as follows:

1) User uploads a single retinal images to the interface
2) The user will click upload, and the interface will preform the analysis
3) Once the analysis is complete, the interface will display two items:
   - Predicted diagnostic grade (eg: Predicted Grade - 2 (moderate))
   - A generated explanation of why that grade was given via features in the image

This implementation is simple to use, and does not have sharp learning curve, it is the same as uploading an image to any other software. The UI clearly displays the results, and is simple to understand and follow for both the healthcare provider and for any patient.

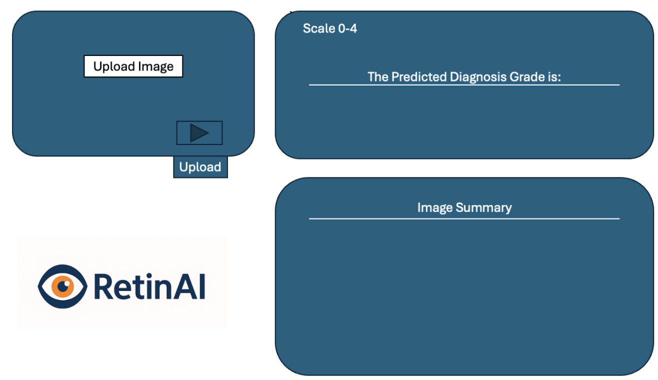


Fig. 4. Sample Gradio Interface

The current stage of the implementation is as follows. There is a Gradio interface that uses the current IDRID trained LLaVA model. The interface consists of multiple sections. The user will upload an image and the model will classify that image and provide some information based off of the grade. If the image is not visible or there is no detection, the output will ask the user to upload a different image.

1) Description of grades and meanings
2) Image upload section
3) Instructions section
4) Predicted grade
5) Recommendation based off of grade
6) Output of model and description of image

The current interface is in basic design stage, it will be improved visually later with the logo and other features at a later date.

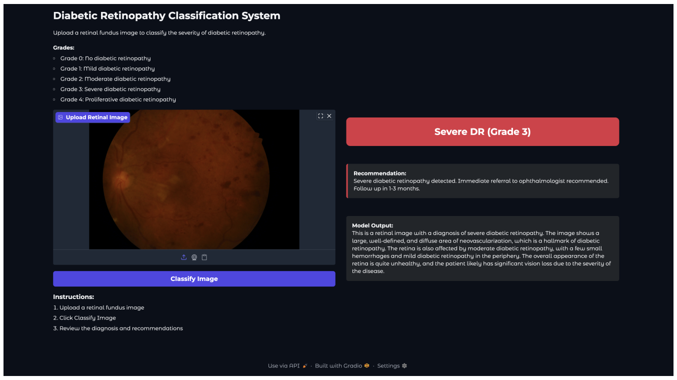


Fig. 5. Current Gradio Interface

V. Results

Currently the model is only trained on the IDRID dataset, and as a result the performance is not extremely high. However, this is important as we do not want the model to fully learn the IDRID dataset, it is purely just for initially training the LLaVA model on Diabetic Retinopathy data.

In the future, the model will be trained on the APTOS dataset, which is significanly larger than the IDRID dataset, and will have higher constraints for the performance. The current performance is as follows.

TABLE I
Baseline Model Performance on IDRiD Test Set

Class	precision	recall	f1-score	support
No DR	0.94	0.88	0.91	34
Mild DR	0.29	0.40	0.33	5
Moderate DR	0.84	0.84	0.84	32
Severe DR	0.79	0.79	0.79	19
Proliferative DR	0.85	0.85	0.85	13
accuracy			0.83	103
macro avg	0.74	0.75	0.74	103
weighted avg	0.83	0.83	0.83	103

The initial fine-tuning on the 413 image IDRiD dataset gives a promising baseline accuracy of 82.52%. Critically, the model produced 103 out of 103 valid predictions, demonstrating that it successfully learned the classification task on this small dataset. This is a bit concerning as overfitting is an issue that may be faced.

The classification report shows strong performance on the easiest No Diabetic Retinopathy and most severe cases. As expected, the model struggled most with the more subtle Mild DR, which is a topic that will hopefully be adressed via the APTOS dataset.

These results are excellent for a baseline model. They confirm that the model is learning the correct visual features of diabetic retinopathy and is not just simply overfitting. This successfully trained model now serves as an ideal starting point for the transfer learning on the larger and more diverse APTOS dataset.

VI. Challenges

This is a unique project as it is using a VLM instead of a traditional classification model, which can only classify the images based on the grade scale. This project aims to continue from that, and also provide a description of why that diagnosis was given to that image.

An important challenge will be the high computational cost for this project. This will require the use of a powerful computer, such as Hipergator to do the training. The APTOS dataset training is the next step and will involve planning a well formatted SLURM request.

Another issue is from the way VLMs work, and the fact that they can sometimes generate false text, that may not be true. This is an issue due to the nature of this project. As a medical project, ensuring that the model is outputting correct information is of the highest priority. If this information is going to be used in a medical setting, patients must be under the impression that the model is preforming correctly, and is actually giving correct diagnosis, in order to assist the healthcare provider. Luckly due to the amount of additional information in the IDRiD dataset regarding pixel level annotations, this will hopefully not be an issue, as the model can be trained specifically to look for these fine features.

The next steps before deliverable 3 are to finish optimizing the IDRID training, improving the Gradio UI, and begin the APTOS training and data formatting. The dataset is unbalanced and data augmentation is necessary to increase the amount of images in certain Retinopathy grades.

VII. AI Reflection

This project is using a VLM, this is not a replacement for a human and cannot be the sole use in a diagnosis. The model is purely a check and second opinion for the user. The model is trained on limited data, and cannot always be generalized to all populations. As a result the model may not preform well on populations not highly represented in the IDRiD and APTOS datasets.

In terms of environmental concerns, training these large models is very energy consuming, and utilizes a lot of resources, however by using a smaller dataset to pretrain, this will hopefully be limited.

VIII. Timeline

TABLE II
PROJECT TIMELINE AND EXPECTED OUTCOMES

Week	**Focus**	**Expected Outcome**
Oct 21 – Oct 28	Data Preprocessing & Baseline Setup	Working data loaders for both APTOS and IDRiD datasets. Completed EDA.
Oct 29 – Nov 5	Initial Model Training & UI Prototype	LLaVA fine tuning on the IDRiD dataset. Basic Gradio UI prototype is built.
Nov 6 – Nov 19	Full Training & Evaluation	Model fine tuned on the APTOS dataset. Initial performance metrics established.
Nov 20 – Nov 27	Model Tuning & UI Integration	Model performance is improved through hyperparameter tuning and refinement of text prompts. The best model is integrated into the Gradio UI for a functional prototype.
Nov 28 – Dec 1	Final Testing, Demo Prep & Report	testing of the application is complete. The final presentation and technical report are prepared and finalized.