tazman02 / **Analysis-of-the-Results-of-Image-Data-Segmentation-Methods** Public

Notifications Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

main 1 Branch 0 Tags Go to file Code

tazman02 added documnetation v2 2f8a326 · now 34 Commits

- gui
- img
- lib
- Github_documentation.pdf
- README.md
- class_names.json
- howto.txt
- ico.png
- out.json
- requirements.txt
- segmentation_app.py
- testy.ipynb

About

No description, website, or topics provided.

Readme
Activity
0 stars
1 watching
0 forks
Report repository

Releases

No releases published

Packages

No packages published

Languages

Language	Share
Python	84.3%
Jupyter Notebook	15.7%

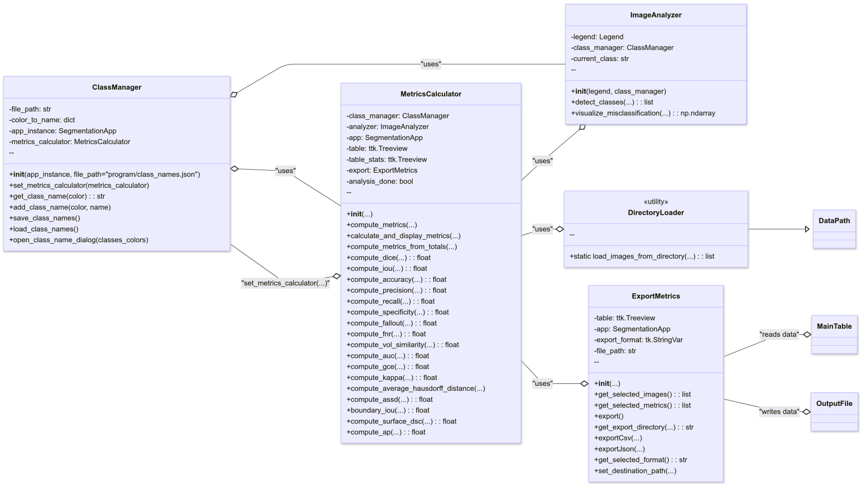
README

Analysis-of-the-Results-of-Image-Data-Segmentation-Methods

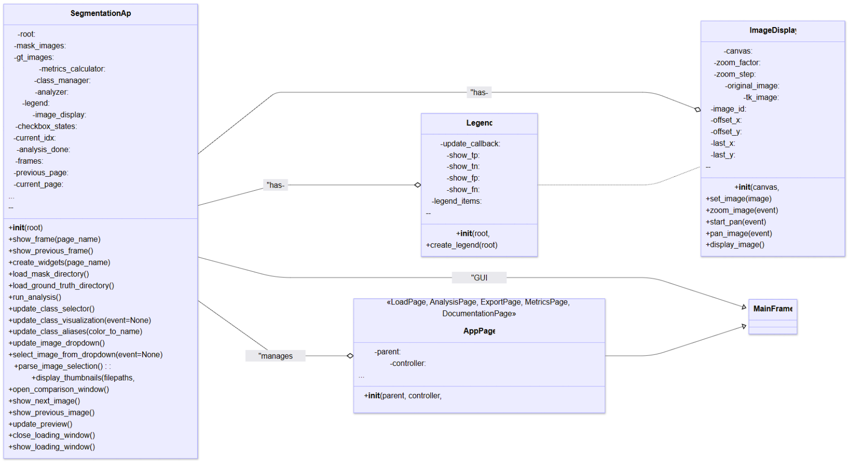
Specification of the requirements

Thesis goal: Develop Python application that can analyze results of image segmentation using selected metrics. Application allows visualization, comparison and export of these metrics.
Target audience: students, researchers, developers working with image segmentation models
Requirements:

- Input: two folders with images (ground truth and prediction)
- Functionalities: calculation of metrics, visualization of classification (TP, FP, TN, FN), comparison of the results, export into CSV/JSON
- Other: simple GUI, scalability, expandability
  Solution proposal: Architecture: Application is split into modules:
- Data loading (DirectoryLoader)
- Class management (ClassManager)
- Metric calculation (MetricsCalculator)
- Visualization and export (ImageAnalyzer, ExportMetrics) UML diagrams: Function diagram – classes and their relations


ImageAnalyzer
-legend: Legend
-class_manager: ClassManager
-current_class: str
--
+init(legend, class_manager)
+detect_classes(...) : : list
+visualize_misclassification(...) : : np.ndarray
"uses"
ClassManager
-file_path: str
-color_to_name: dict
-app_instance: SegmentationApp
-metrics_calculator: MetricsCalculator
--
+init(app_instance, file_path="program/class_names.json")
+set_metrics_calculator(metrics_calculator)
+get_class_name(color) : : str
+add_class_name(color, name)
+save_class_names()
+load_class_names()
+open_class_name_dialog(classes_colors)
MetricsCalculator
-class_manager: ClassManager
-analyzer: ImageAnalyzer
-app: SegmentationApp
-table: ttk.Treeview
-table_stats: ttk.Treeview
-export: ExportMetrics
-analysis_done: bool
--
+init(...)
+compute_metrics(...)
+calculate_and_display_metrics(...)
+compute_metrics_from_totals(...)
+compute_dice(...) : : float
+compute_iou(...) : : float
+compute_accuracy(...) : : float
+compute_precision(...) : : float
+compute_recall(...) : : float
+compute_specificity(...) : : float
+compute_fallout(...) : : float
+compute_fnr(...) : : float
+compute_vol_similarity(...) : : float
+compute_auc(...) : : float
+compute_gce(...) : : float
+compute_kappa(...) : : float
+compute_average_hausdorff_distance(...)
+compute_assd(...) : : float
+boundary_iou(...) : : float
+compute_surface_dsc(...) : : float
+compute_ap(...) : : float
"uses"
"uses"
"set_metrics_calculator(...)"
"uses"
«utility»
DirectoryLoader
--
+static load_images_from_directory(...) : : list
DataPath
"uses"
ExportMetrics
-table: ttk.Treeview
-app: SegmentationApp
-export_format: tk.StringVar
-file_path: str
--
+init(...)
+get_selected_images() : : list
+get_selected_metrics() : : list
+export()
+get_export_directory(...) : : str
+exportCsv(...)
+exportJson(...)
+get_selected_format() : : str
+set_destination_path(...)
"reads data"
MainTable
"writes data"
OutputFile

User diagram – UI components and their logic



GUI design:

- Navigation panel (LOAD -> ANALYSIS -> METRICS -> EXPORT)
- Sections: Loading, Analysis, Metrics, Export

Realization and implementation

Used technologies:

- Python 3.12
- OpenCV, NumPy, SciPy, Tkinter
- Git, GitHub, Figma, TkForge Implementation:
- Calculation of metrics: functions compute_<metric_name>()
- GUI: navigation panel, interactive controls (dropdowns, checkboxes ...)
- Export: preview and format selection (CSV/JSON)

Testing and verification of the results

Data for testing:

- Simple synthetic segmentation dataset (ChatGPT) Comparison with reference values:
- Verification of metric values with existing libraries like sklearn and medpy
- Table of compared values: majority of metrics are equivalent, some variation explained in thesis Limits:
- Synthetic data results don’t have to apply in all real-world scenarios
- Some metrics don’t have publicly accessible implementation

Installation, usage and maintenance

Installation:

- Install supported version of Python, currently 3.12 (https://www.python.org/downloads/)
- Run `git clone https://github.com/tazman02/Analysis-of-the-Results-of-Image-Data-Segmentation-Methods` to fetch the latest version of the project
- Install libraries from requirements.txt (Run: `pip install -r requirements.txt`)

- Run the main Python file from the project directory (Run: `python segmentation_app.py`) Usage:
- Load ground truth and prediction datasets (WARNING: Path cannot contain any diacritics, may cause error)
- Optional: Set class aliases for clarity in metrics
- Run the analysis module to compute evaluation results
- View results via the UI or console output
- Export computed metrics to desired format (e.g., CSV, JSON) Maintenance:
- Pull the latest version from GitHub (Run: `git pull origin main` or your current working branch)