IMAGE PROCESSING TOOL

A WPF APPLICATION FOR LOADING, TRANSFORMING, AND SAVING IMAGES FROM A LOCAL FOLDER.
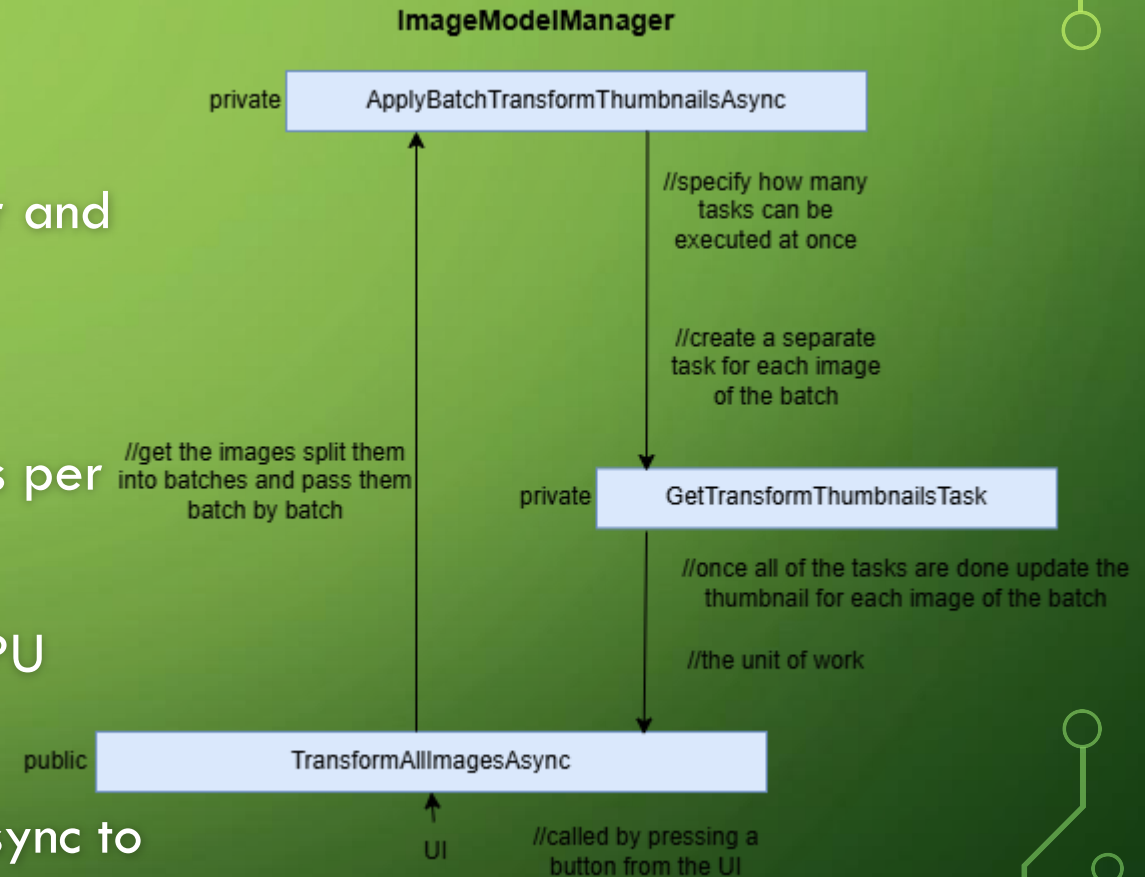
Authors:

Miroslav Andonov

Yana Ivanova

PROJECT OVERVIEW

- Built with .NET WPF, utilizes asynchronous and parallel programming.

- Handles over 500 images efficiently (dataset: 716 bird images)

- Core features:
  - Resize thumbnails (e.g., 128×128 → 64×64)
  - Convert to grayscale
  - Undo last 4 actions
  - Save to custom folder
  - Search/filter by name
  - Cancel long operations
  - Configure batch size

PROCESSING PIPELINE

- Core logic managed by ImageModelManager and its subclasses:
  - Load, Grayscale, Resize, and Save Managers
- Batching limits memory usage (e.g., 32 images per batch)
- Use Semaphore to control concurrency (2 x CPU cores)
- Thumbnails updated using Dispatcher.InvokeAsync to remain UI-thread safe

ImageModelManager

private ApplyBatchTransformThumbnailsAsync

//specify how many tasks can be executed at once

//create a separate task for each image of the batch

//get the images split them into batches and pass them batch by batch

private GetTransformThumbnailsTask

//once all of the tasks are done update the thumbnail for each image of the batch

//the unit of work

public TransformAllImagesAsync

UI

//called by pressing a button from the UI

MODELS, CONTROLS AND SEARCH LOGIC

- ImageModel represents image state (name, path, bitmap, visibility)

- Bound to ThumbnailControl UserControl with UI thread safety

- Implements INotifyPropertyChanged for real-time UI updates

- Search uses AsParallel() for responsive filtering

- Batch size configurable via the Settings window and appsettings.json