Artwork Super-resolution Scanning Application

ECpE SENIOR DESIGN – GROUP 18

Reece Dodge, Isaac Plambeck, Garrett Powell, Samuel Schaphorst

Advisor/Client: Thomas Daniels

PROJECT OVERVIEW

THE PROBLEM

In today's world, freelance artists, print shops, museums, and art galleries have no way of cheaply/easily producing a true-to-life digital version of physical artwork.

OUR SOLUTION

Create an application that uses a set of 3-10 user-uploaded photos of a single canvas and automatically produces a digital version according to a target output resolution.

TARGET USERS

- Freelance Artists
- Print Shops
- Art Galleries
- Museums

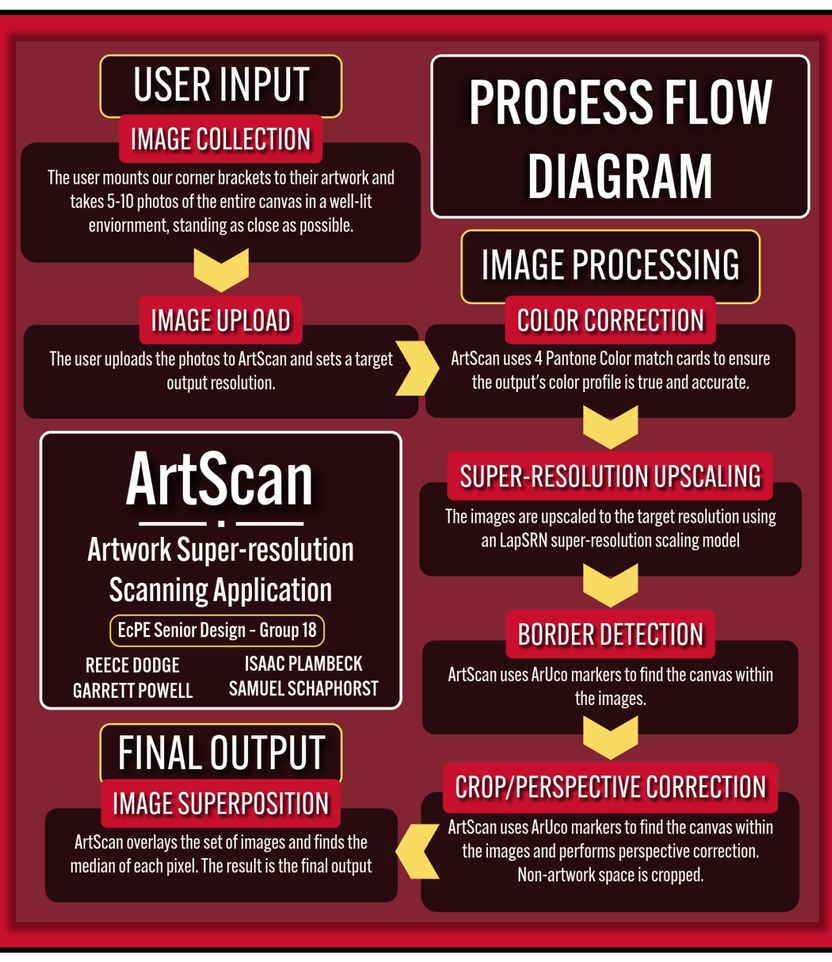
REQUIREMENTS

- Application based on Python/OpenCV
- User-friendly UI
- Lightweight image processing algorithms
- Ability to output 300+ DPI images

SPECIFICATIONS

- OpenCV version 4.8
- Pantone Color Correction
- LapSRN Super-resolution

PROCESS FLOW DIAGRAM



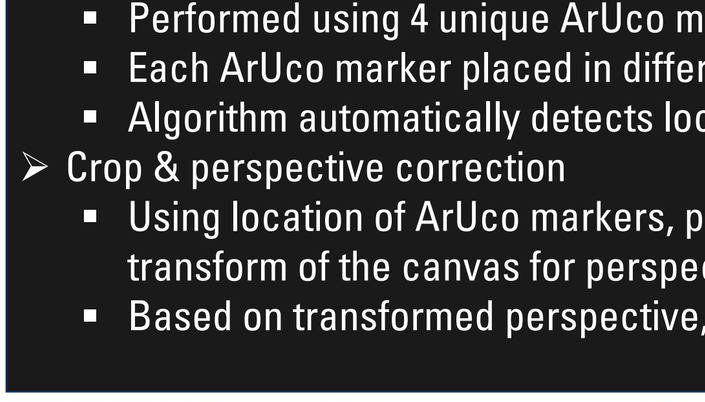
OUR IMPLEMENTATION

METHODOLOGY

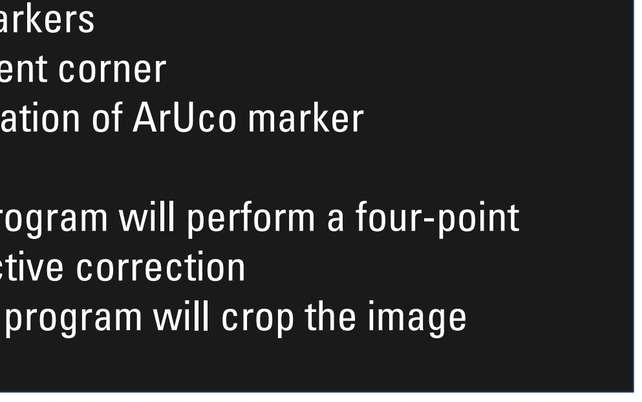
- Image collection through wooden frames attached to corner of canvas
  - Each frame equipped with Pantone card and ArUco marker
- Image processing performed automatically through application to create super-resolution output
- Process flow:
  - Load a set of images
  - Image resizing
  - Color correction
  - Corner detection
  - Crop & perspective correction
  - Apply median to all images

TESTING

- Integration Testing
  - Testing of cropping and color changing algorithms with multiple different images
- Acceptance Testing
  - Feedback from client during Meetings
- Unit Testing
  - Used to Test function parameters with images sizes and amounts



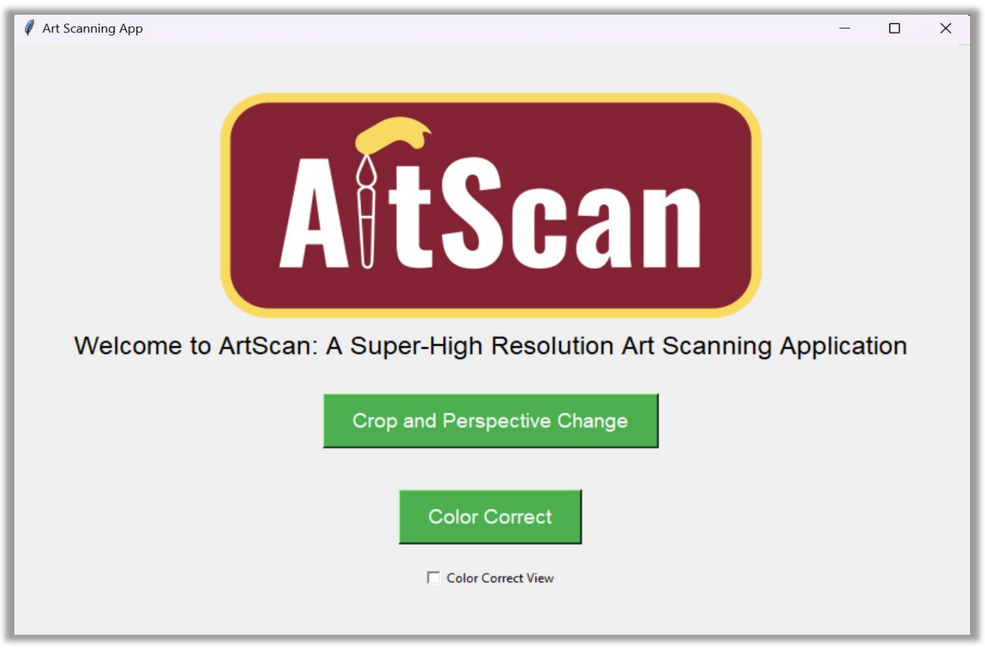
EXAMPLE INPUT



OUTPUT FILE

IMAGE PROCESSING STEPS

- Color correction
  - Performed using 4 Pantone cards
  - Each card placed in different corner
  - Correction contributions at each pixel is based on distance to each card
  - Reference image of Pantone in ideal lighting conditions used to comparatively adjust color histograms of input images
- Corner detection
  - Performed using 4 unique ArUco markers
  - Each ArUco marker placed in different corner
  - Algorithm automatically detects location of ArUco marker
- Crop & perspective correction
  - Using location of ArUco markers, program will perform a four-point transform of the canvas for perspective correction
  - Based on transformed perspective, program will crop the image



COLOR REFERENCE

TEST INPUT IMAGE

COLOR CORRECTED OUTPUT

RESULTS

CONCLUSION

- Our program provides a convenient way to digitalize artwork
- Program utilizes Python and OpenCV to efficiently produce high resolution image
- Use of pantone color cards and ArUco markers allow program to color correct and border detect
- Wooden brackets and Velcro patches allow for collection of data on mounted canvases
- Program is intended to produce results in a quick efficient manner

SUMMARY

- Upload multiple images to software, click the start button, and wait for finalized output
- Provides faster method of processing multiple images

IMPACT

- Supports artists
- Encourages artists to digitalize artwork
- Prevents loss of original work